Dokumentation für yaVDR

Inhaltsverzeichnis

Widmung

Für das yaVDR-Team, das mir den VDR näher gebracht hat und alle fleißigen Entwickler und Nutzer, die mir weitergeholfen haben.

Vorwort



Diese Dokumentation soll die Bedienung und Interna eines yaVDR-Systems erklären und dem Nutzer dabei helfen, sich mit den Besonderheiten, die yaVDR von einem normalen Ubuntu unterscheiden, vertraut zu machen. Es ist bewusst keine Einführung in die Grundlagen der Bedienung von (Ubuntu) Linux. Daher wird dem Neueinsteiger empfohlen sich mit Hilfe von entsprechender Literatur oder Online-Quellen wie dem sehr umfangreichen Ubuntu Users Wiki einzulesen. Auch für den VDR und für XBMC kann an dieser Stelle nicht jedes Detail erklärt werden - gute Anlaufstellen für weiterführende Informationen sind das VDR-Wiki und das XBMC-Wiki

Wann immer die Beschreibung von Befehlen, Funktionen oder Abläufen den Rahmen dieser Dokumentation sprengen würde, wird daher auf entsprechende Artikel verlinkt. Auch doppelte Ausführungen zum selben Themas werden wo immer es geht vermieden und stattdessen auf die entsprechende Stelle in der Dokumentation verlinkt.

Diese Dokumentation gibt es auch als PDF- und als "All-in-one" Version

Lizenz

Diese Dokumentation steht unter der GNU FDL [1]

Konventionen

Befehle, die man in der Shell oder in xterm ausführen kann, sehen so aus:

Ein Beispiel-Befehl

```
❶user@yavdr:~$ ❷ sudo make me a sandwich
```

❶ Der Shell-Prompt zeigt, dass man als Benutzer **user** am Rechner **yavdr** angemeldet ist und sich im $HOME Verzeichnis [2] befindet.

❷ dahinter steht der Befehl, der Ausgeführt werden soll [3]

[1] Free DOC LICENCE, siehe http://www.gnu.org/copyleft/fdl.html

[2] Das $HOME-Verzeichnis ist das individuelle Benutzerverzeichnis auf Linux-Systemen. Es befindet sich normalerweise unter */home/Benutzername/* und wird durch das Tilde-Zeichen ~ abgekürzt.

[3] das Beispiel orientiert sich an http://xkcd.com/149/ und soll das Konzept des Superusers (siehe http://wiki.ubuntuusers.de/-sudo) verdeutlichen.

Anmerkungen zu bestimmten Punkten werden als mit Nummern versehene kleine schwarze Kreise kenntlich gemacht.

Konfigurationsdateien werden mit Source-Code-Highlighting dargestellt:

Beispiel für die Konfigurationsdatei /etc/fstab Über diese Datei kann man Linux anweisen Speichergeräte fest in das Root-Dateisystem einzuhängen

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sdb1 during installation
UUID=83a8163c-195b-4e85-a9ed-18478339dc3f /               ext4    errors=remount-ro 0  ↩
          1
# swap was on /dev/sdb5 during installation
UUID=32fc4d86-ab86-4ed8-bf24-68aa98ec6d0c none            swap    sw              0       0
```

Tipps
Tips werden durch dieses Symbol gekennzeichnet

Notizen
Notizen werden durch dieses Symbol gekennzeichnet



Achtung!
Hinweise werden durch dieses Symbol gekennzeichnet



Wichtig!
Wichtiges wird durch dieses Symbol gekennzeichnet



Warnung!
Warnungen werden durch dieses Symbol gekennzeichnet

Anmerkung

Anmerkungen erscheinen auf gelbem Hintergrund.

Kapitel 1

yaVDR

1.1 Einleitung

yaVDR [1] ist eine auf Ubuntu basierende Linux-Distribution, die speziell für die Nutzung des Video Disk Recorder von Klaus Schmidinger angepasst wurde. Sie ist bestens geeignet um digitales und analoges [2] Fernsehen über alle gängigen Empfangswege [3] zu empfangen, aufzuzeichnen, zu bearbeiten und wiederzugeben.

Mit XBMC ist außerdem ein mächtiges MediaCenter mit an Bord, das es ermöglicht einen PC mit yaVDR als praktisch vollwertigen HTPC [4] zu nutzen.

Um neue Funktionen schnell zu integrieren und die Hardware-Unterstützung zu verbessern setzt yaVDR auf relativ aktuelle (aber dennoch möglichst stabile) Entwicklungs-Versionen.

Um dem Nutzer möglichst viel Konfigurationsarbeit abzunehmen, wird unterstützte Hardware bei der Installation automatisch erkannt und eingerichtet.

Ein Großteil der Einstellungen für das System kann über ein komfortables Webinterface vorgenommen werden.

1.2 Hardware-Voraussetzungen

yaVDR unterstützt ein breites Spektrum von TV-Karten, Fernbedienungen und VDPAU-fähigen nVidia-Grafikkarten.

Seit Version 0.4 wird yaVDR nur noch als 64-Bit Version angeboten, so dass ein entsprechender Prozessor [5] verwendet werden muss.

Die empfohlene Festplattengröße hängt von der gewünschten Aufnahmekapazität ab. Eine Partition für Standard-Installation von yaVDR sollte mindestens 4 GB groß sein, um genug Platz für Updates, XBMC-Metadateien und zusätzliche Pakete zu haben, werden mindestens 6 GB empfohlen. Die Größe der SWAP-Partition entspricht normalerweise der Größe des Arbeitsspeichers. Für SD-TV werden fallen in abhängig von der Bitrate des Senders bis zu 2,7 GB pro Stunde Aufnahme an, für HD-TV bis zu 9 GB pro Stunde. Im normalen Betrieb hat sich eine Festplattengröße von mindestens 160 GB bewährt, für den Fall, das Aufnahmen gesammelt werden sind (aus den Erfahrungen des Autors) 2 TB oder mehr empfohlen.

Weitergehende Fragen versucht die Seite Hardware-Voraussetzungen zu beantworten.

[1] yaVDR steht für yet another VDR
[2] setzt Hauppauge PVR-TV-Karten voraus, siehe PVRinput-Plugin
[3] DVB-S(2), DVB-C, DVB-T
[4] Home Theater PC
[5] AMD Athlon64 oder höher, Intel-Prozessoren mit Intel64-Unterstützung



Checkliste vor Inbetriebnahme

- VDPAU-fähige nVidia-Grafikkarte oder anderes unterstütztes Ausgabegerät (TT S2-6400, Hauppauge PVR350, SD-FF Karte) verbaut?
- Alle Tuner angeschlossen? (Falls nicht bitte daran denken eine entsprechende **udev-Regel** für das **Dynamite-Plugin** anzulegen!)

TODO: udev-Regeln für dynamite erstellen und verlinken

- Netzwerkkabel eingesteckt?

1.3 Installation von yaVDR

1.3.1 Download

Die aktuelle stabile Version von yaVDR steht im Download-Bereich der offiziellen Website bereit.

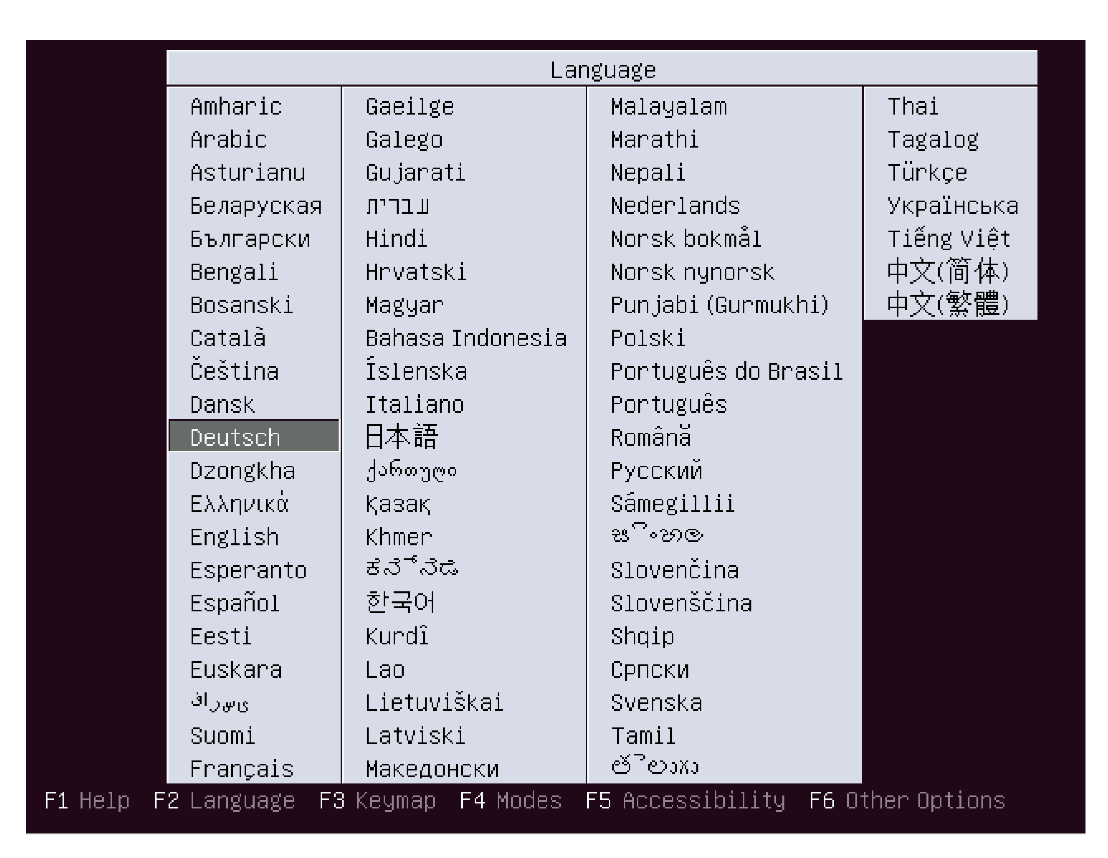
1.3.2 Installation

Je nach verwendetem Prozessor, Arbeitsspeicher und Festplatten nimmt eine yaVDR-Installation zwischen 20 bis 70 Minuten in Anspruch. Der PC, auf dem yaVDR Installiert werden soll, muss nun so eingestellt werden, dass er von der eingelegten Installations-CD startet. Wie das geht steht normalerweise im Handbuch des Herstellers des Systems bzw. des Mainboards.

Bootmenü
Bei vielen aktuellen Mainboards kann man mit den Tasten F8 oder F12 ein Menü aus der POST-Anzeige [a] heraus aufrufen, das es erlaubt das Bootgerät für den Start zu wählen.

[a] Power On Self-Test

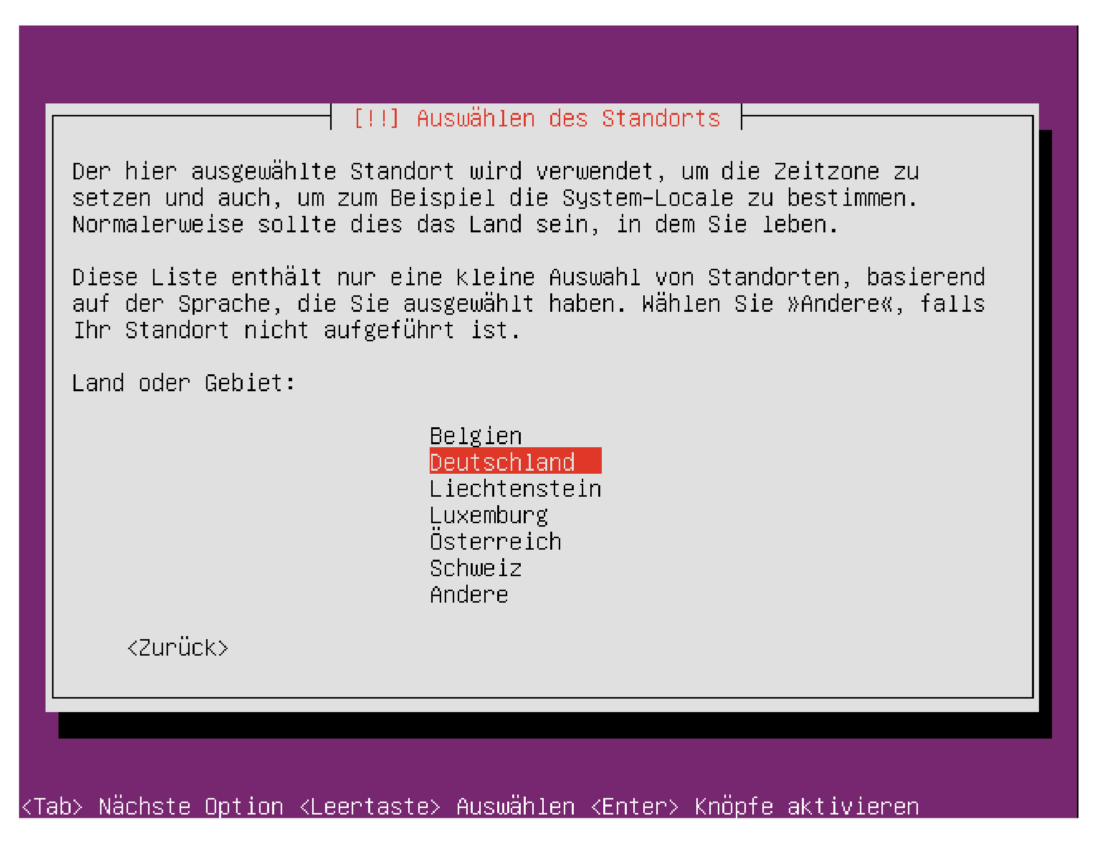
Sobald das System von der Installations-CD bootet, erscheint ein Auswahlmenü in dem man die Sprache für die Installation auswählen kann:



Nun kann man die yaVDR-Installation starten, die CD auf Fehler überprüfen, den Arbeitsspeicher testen, von der ersten Festplatte in der Bootreihenfolge starten oder ein Rettungssystem starten lassen. Um yaVDR zu installieren wählt man also den ersten Punkt in der Liste aus:



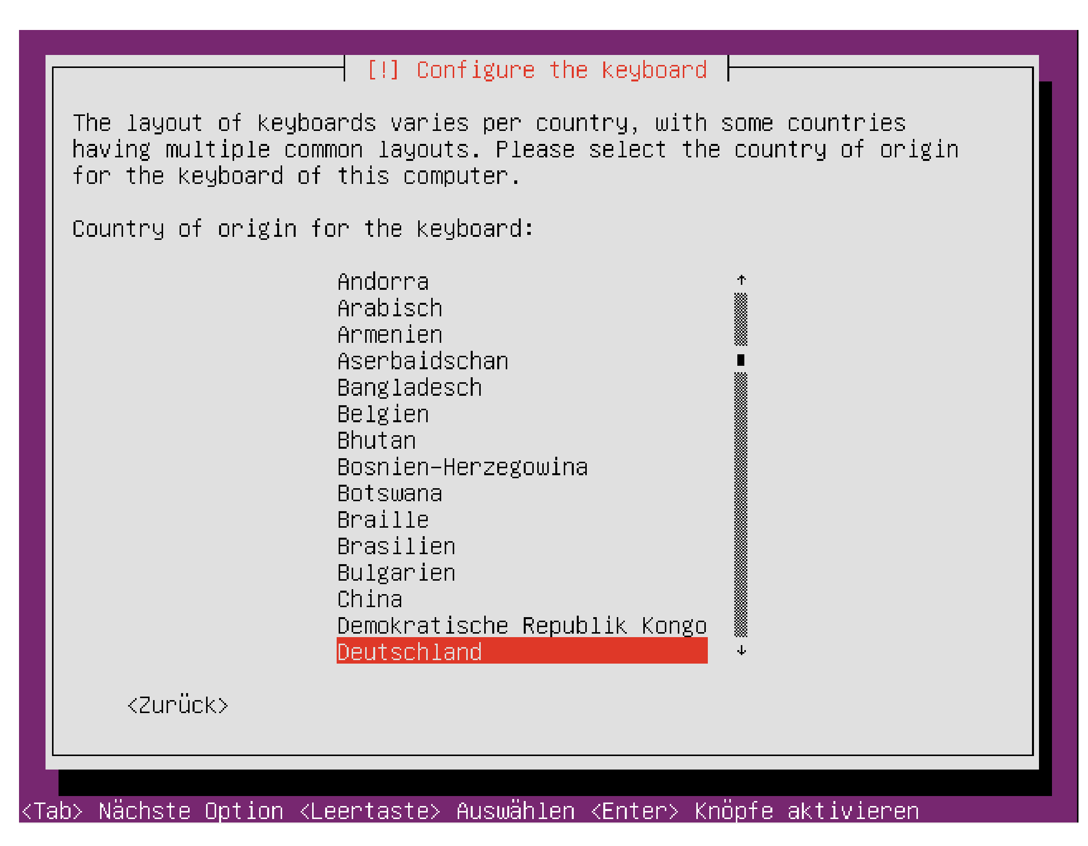
Um Sprache und Zeit auszuwählen wird der Standort des Systems abgefragt:



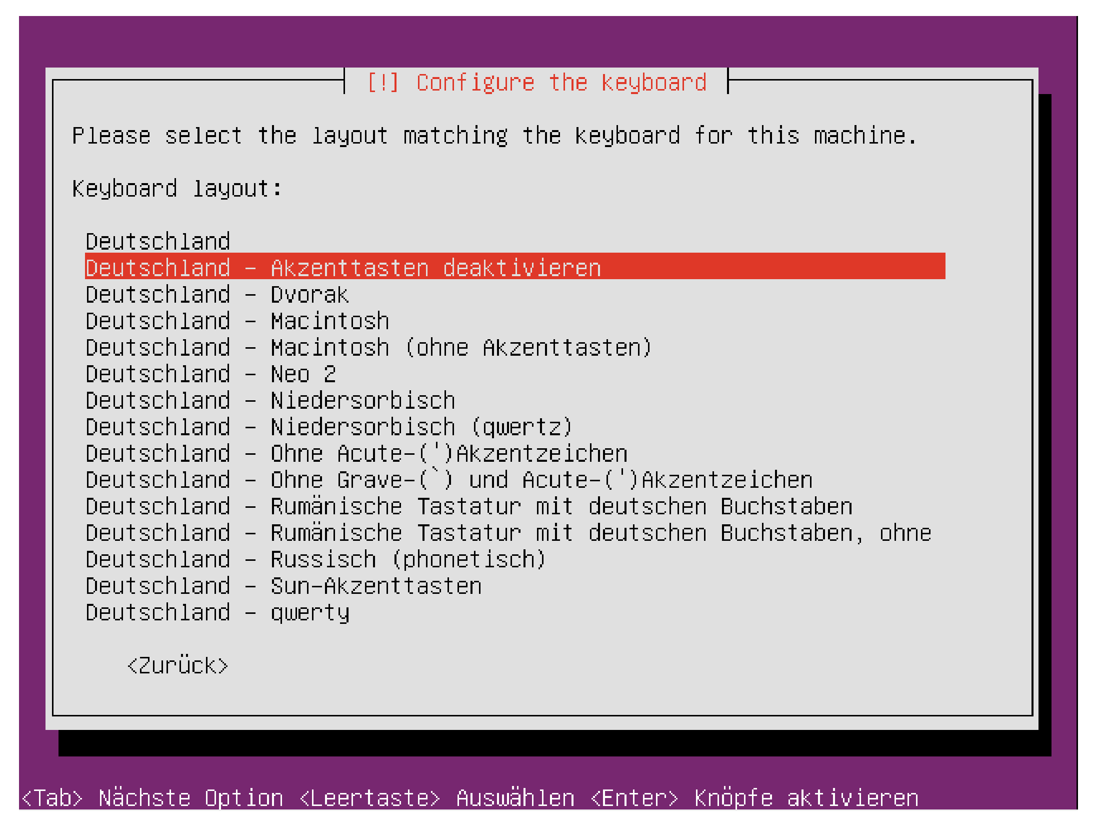
Aufgrund eines Bugs im Installer von Ubuntu 11.04 Natty Narwhal ist es empfehlenswert die Tastenbelegung selbst zu wählen anstatt sie automatisch bestimmen zu lassen:



Als erstes wählt man das Land...



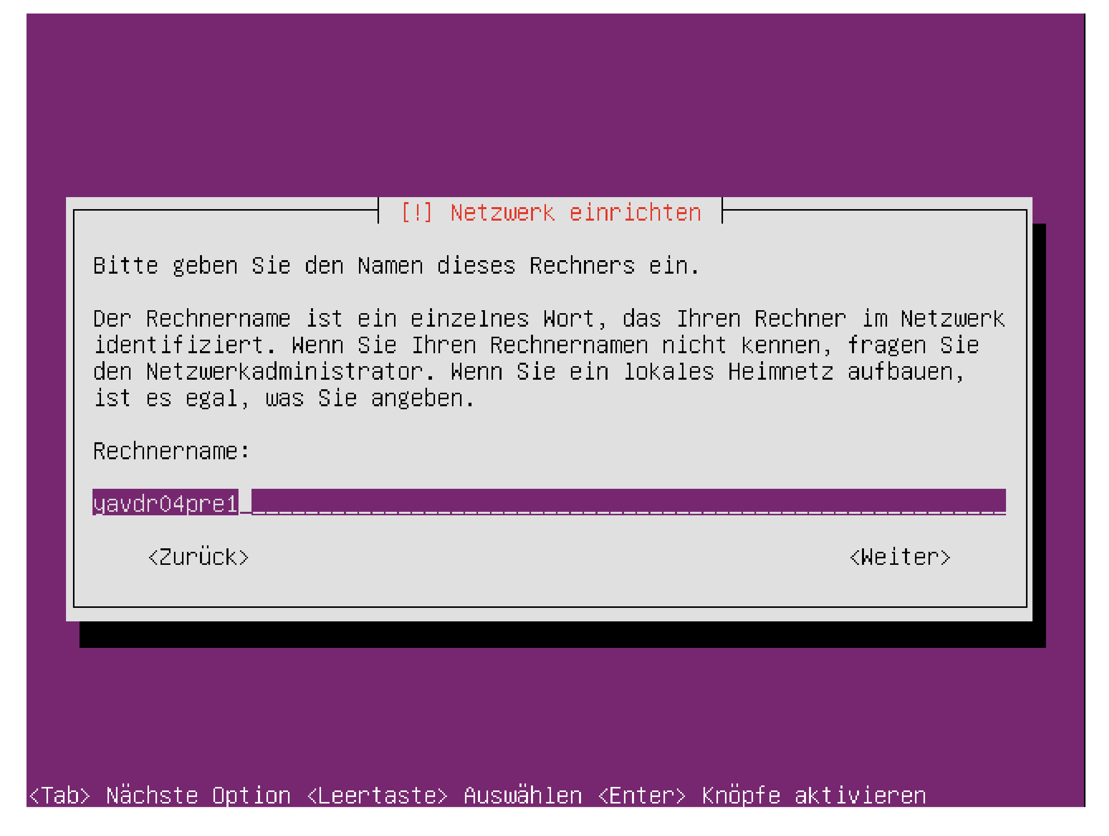
... und danach die Tastenbelegung. Da auf dem yaVDR-System i.d.R. keine literarischen Meisterwerke entstehen sollen, aber ggf. Einstellungen in der Shell vorgenommen werden sollen, bietet es sich an die Akzenttasten zu deaktivieren, so dass die Eingabe von häufig verwendeten Zeichen wie z.B. der Tilde "~" erleichtert wird:



Vor Beginn der weiteren Konfiguration werden einige Dateien nachgeladen:

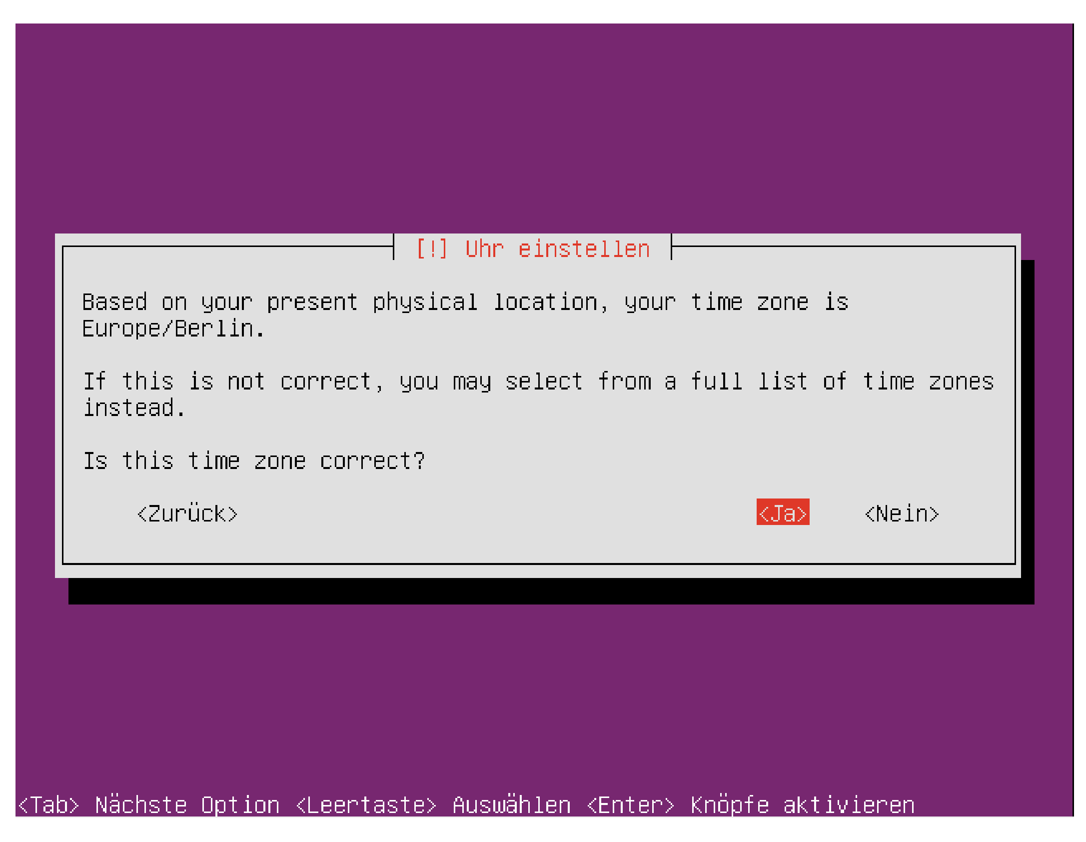


Dann wird versucht das Netzwerk automatisch zu konfigurieren und es kann der Hostname [6] eingestellt werden:



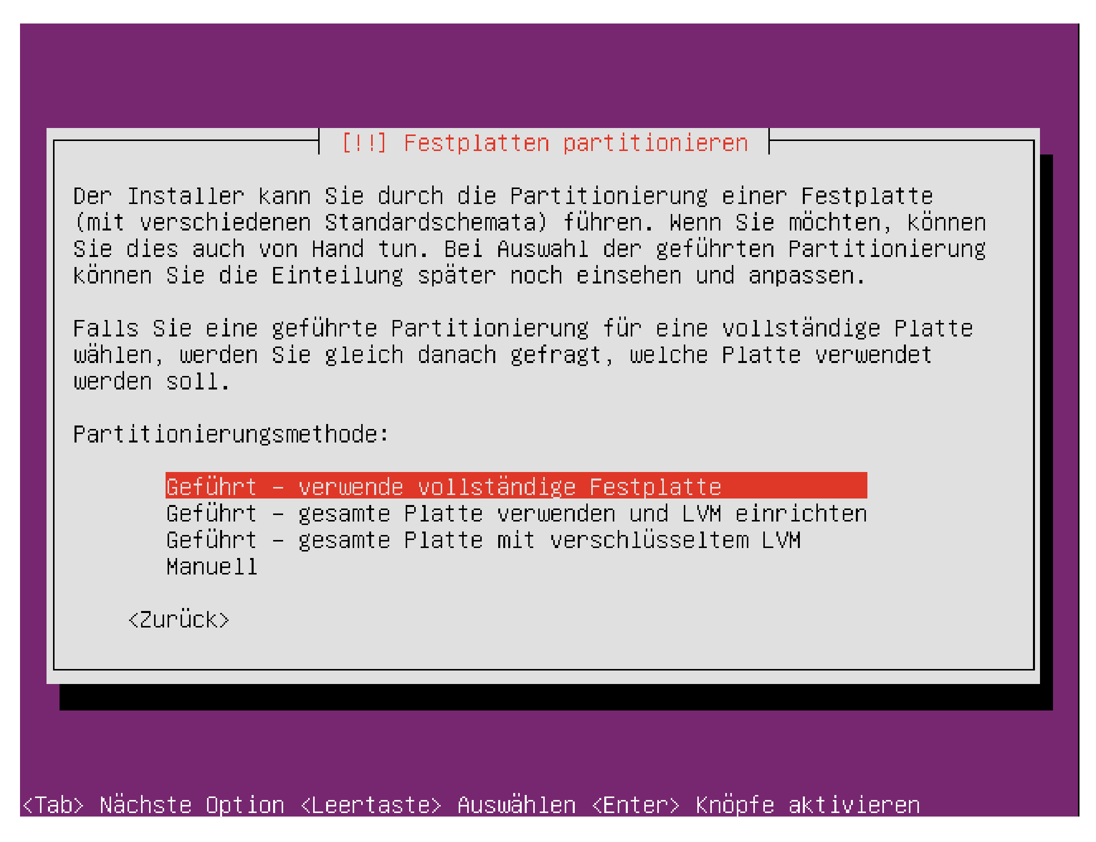
Nun kann die Zeitzone eingestellt werden:

[6] Das ist der Name unter dem der Rechner im Netzwerk ansprechbar ist

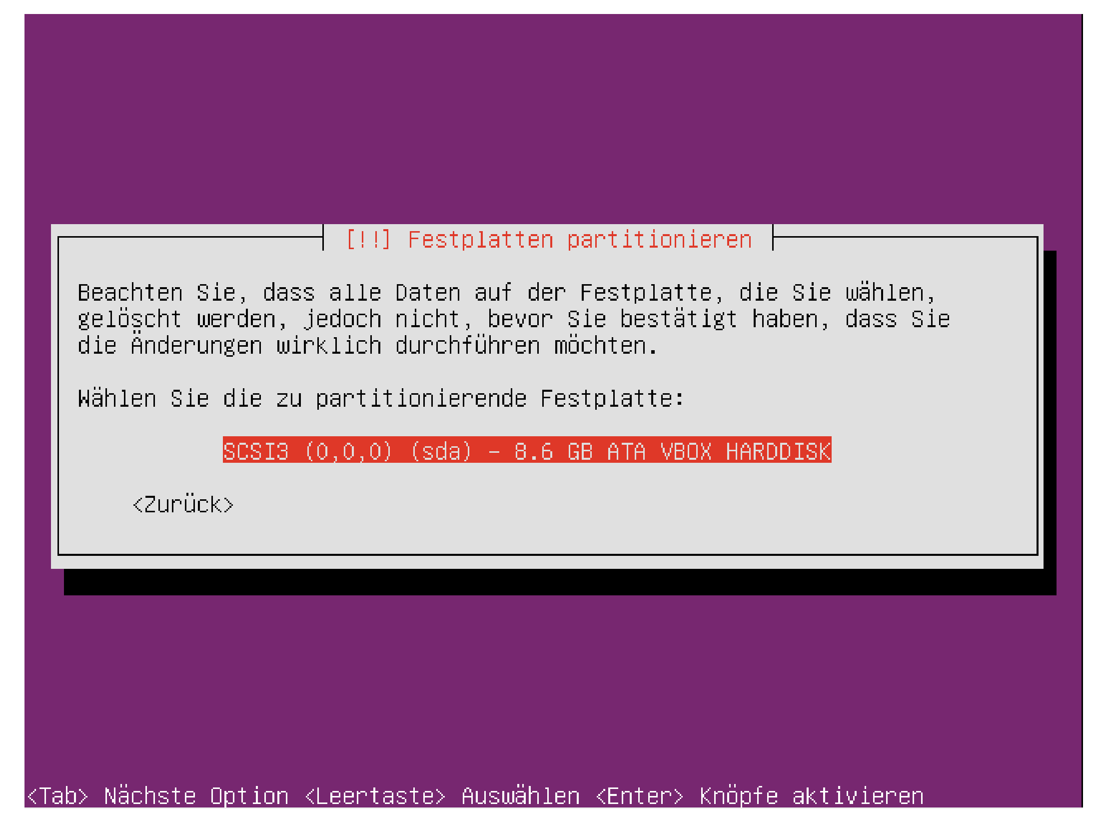


Im nächsten Schritt kann man konfigurieren, wie die verbaute(n) Festplatten genutzt werden sollen. Die Root-Partition sollte mindestens 3 GB groß sein, damit genug Platz für die Installation weiterer Pakete bleibt ist eine Größe von 8 - 20 GB empfehlenswert. Alle lokalen Mediendateien sollten später unter /srv/vdr liegen. Dies ist der bei yaVDR voreingestellte Ort, an dem später Aufnahmen abgelegt werden und an dem XBMC Medien in den entsprechenden Unterverzeichnissen erwartet. Näheres zu dem Thema findet sich im Abschnitt Verzeichnissstruktur von yaVDR.

Daher kann man entweder den Installer automatisch die Festplatte aufteilen lassen ("**Geführt - verwende vollständige Festplatte**"), wodurch neben der SWAP-Partition der restliche Platz für Root verwendet wird (siehe folgender Screenshot) oder manuell eine kleine Partition für **/** erstellen und den Verbleibenen Platz (oder auch eine zweite Festplatte mit hoher Kapazität) unter /srv einhängen. Als Dateisystem eignet sich z.B. ext4.



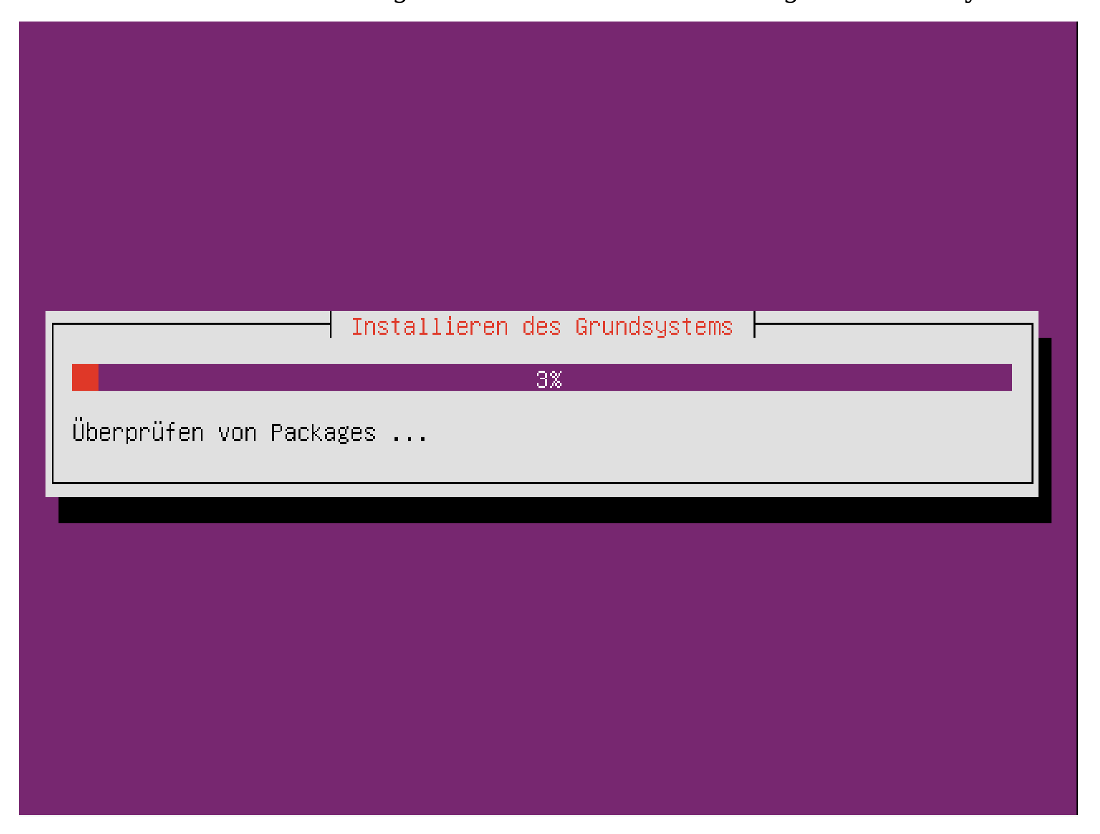
Nun kann die Festplatte ausgewählt werden, auf der das System installiert werden soll:



Und man bestätigt, dass man die Vorgeschlagene Partitionierung übernehmen möchte:



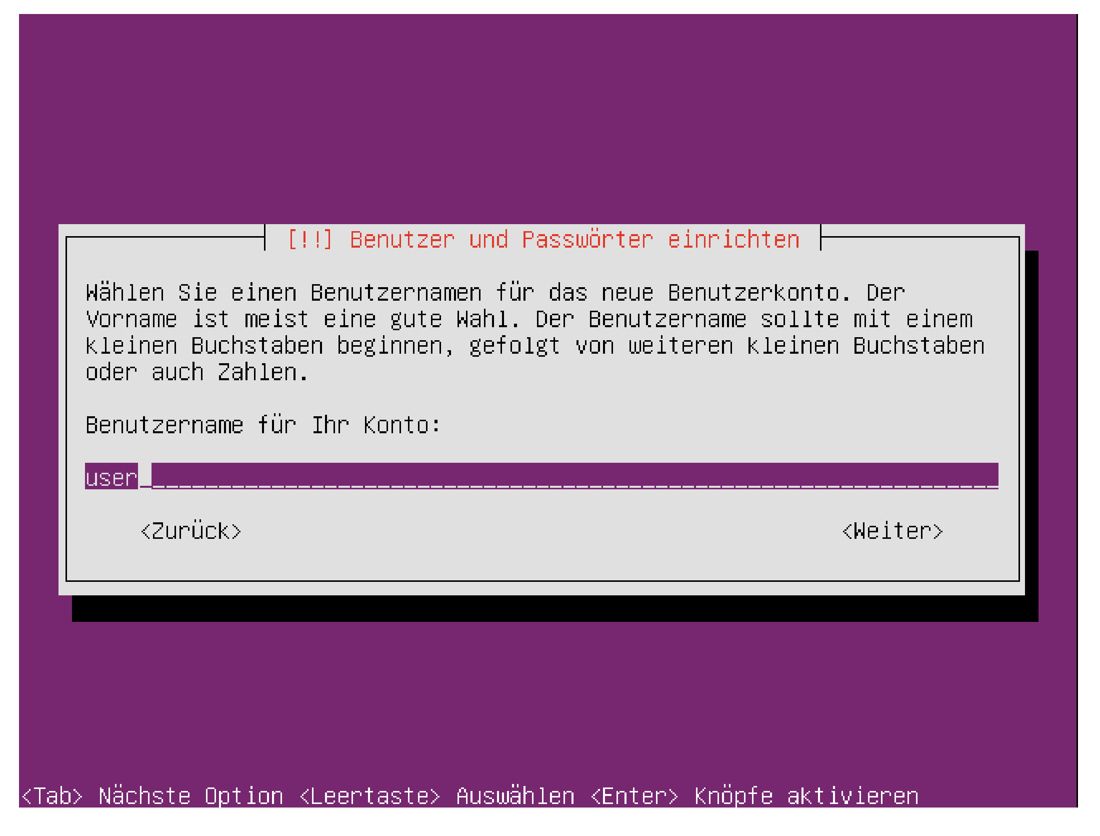
Nun wird die Partitionierung übernommen und damit begonnen das System einzurichten:



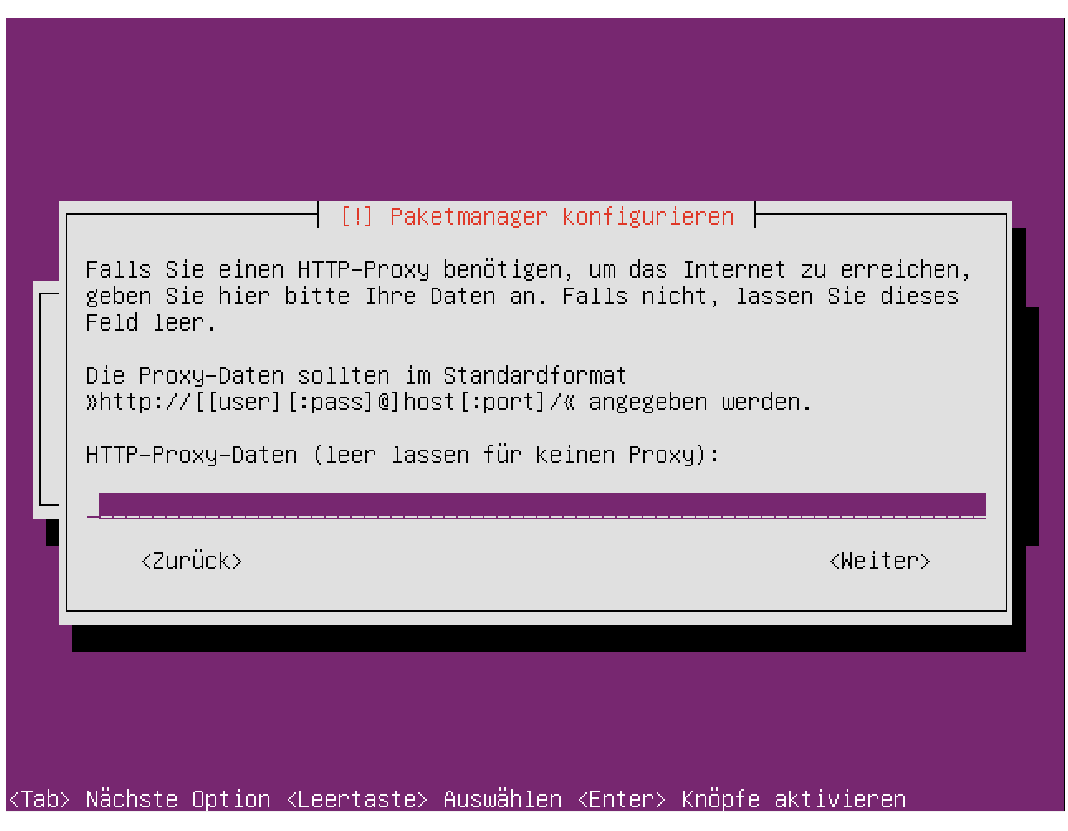
Zum Abschluss der Konfiguration wird noch der Benutzer erstellt, mit dem man sich später zu Wartungs- und Administrationszwecken am yaVDR-System anmelden kann. Dazu gibt man seinen Namen ein:



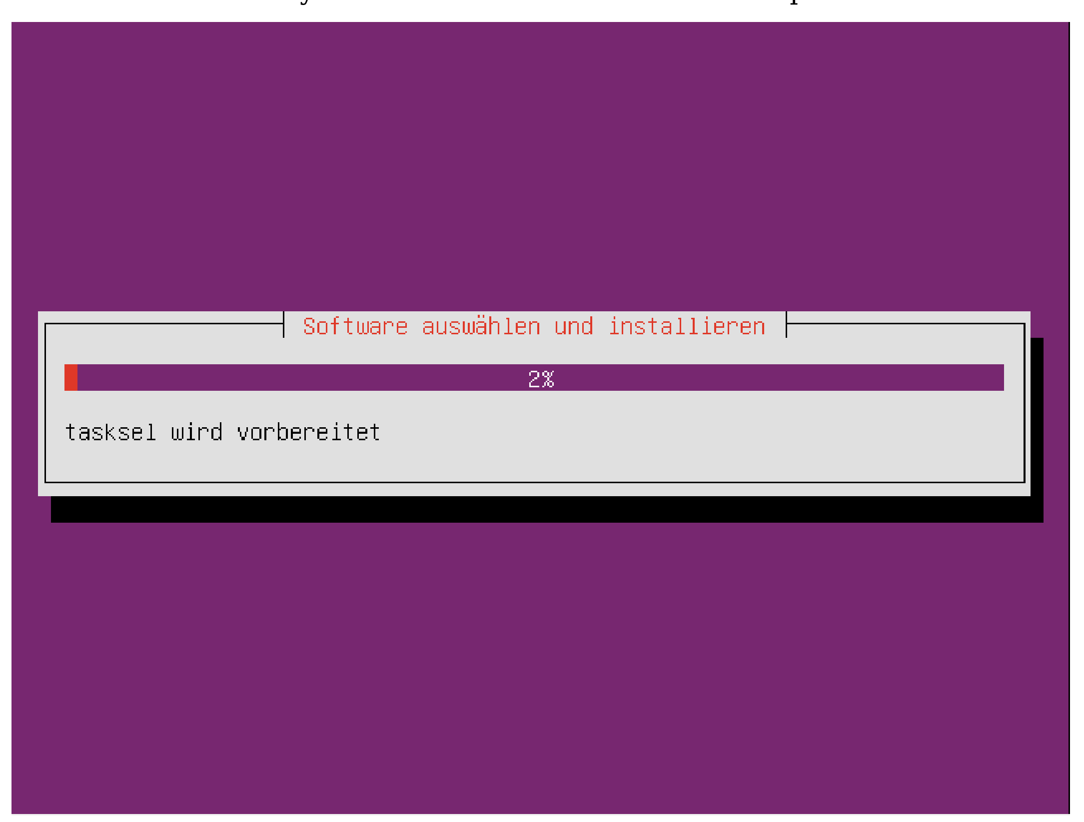
Darauf basierend wird ein Benutzername vorgeschlagen, den man aber nach Belieben ändern kann:



Als vorläufig letzten Schritt muss man noch angeben, ob ein HTTP-Proxy verwendet wird. Ist das (wie in den meisten Heimnetzwerken üblich) nicht der Fall, kann man das Feld einfach leer lassen:



Die Installation des Systems nimmt nun etwas Zeit in Anspruch:

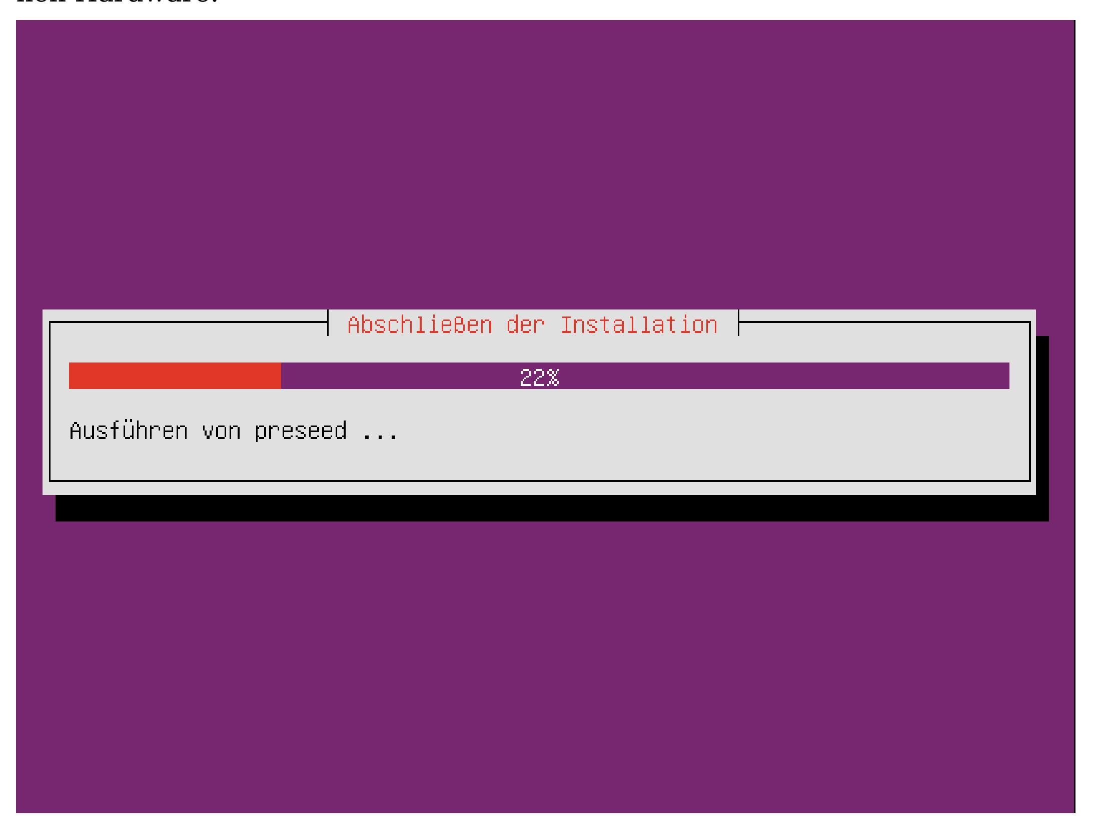


Bevor das System für den ersten Start vorbereitet werden kann, muss noch der Ort für die Installation des Bootloaders gewählt werden. Hat man keine besonderen Wünsche kann man die Standard-Auswahl einfach übernehmen und Grub in den MBR [7] der Systemplatte installieren.

[7] Master Boot Record



Nun erfolgt noch die automatische Konfiguration von yaVDR anhand der bei der Installation gefundenen Hardware.



Ist dieser Prozess abgeschlossen kann man die Installations-CD entnehmen und den PC neu starten.

1.4 Erste Schritte

Nach der Installation startet yaVDR mit drei vorkonfigurierten Internet-TV Kanälen. Ist eine Internetanbindung vorhanden sieht man also zunächst den WebTV-Stream von 3sat.

Erste Schritte nach der Installation
Drei Punkte müssen abgearbeitet werden um yaVDR nach einer Neuinstallation sinnvoll nutzen zu können:

- aktuelle Updates einspielen
- Senderliste erstellen, z.B. mittels Channelpedia im Webfrontend
- Fernbedienung einrichten falls sie nicht direkt nach der Installation wie gewünscht funktioniert

1.4.1 Aktuelle Updates einspielen



yaVDR immer möglichst aktuell halten
yaVDR ist eine "stark entwickelte" Distribution, für die häufig Updates erscheinen. Viele "alte" Fehler und Probleme lassen sich vermeiden wenn man regelmäßig sein System auf den neuesten Stand bringt.
Gerade **bevor** man darüber nachdenkt eine Support-Anfrage im Forum zu stellen, sollte man überprüfen, ob der Fehler auch nach dem Einspielen der aktuellen Updates und einem anschließenden Neustart noch auftritt.

Update des Systems

```
sudo apt-get update && sudo apt-get dist-upgrade
```

1.4.2 Bedienung des VDR

Um die grundlegende Bedienung und die Konfigurationsmöglichkeiten des VDR über das OSD [8] kennenzulernen empfiehlt es sich das Benutzerhandbuch zu lesen: http://vdr-wiki.de/wiki/index.php/Benutzerha

1.4.2.1 Tastaturbelegung

Auch wenn eine Fernbedienung die Bedienung deutlich bequemer macht, ist es nach der Installation oft hilfreich wenn man yaVDR auch mit der Tastatur bedienen kann.

Tabelle 1.1: Tastaturbelegung in yaVDR

Taste	Funktion	Anmerkung
m	**Menü**	
Backspace	**Exit**	Schließt das aktive Menü bzw. kehrt zum Übergeordneten Menü zurück

[8] On Screen Display

Tabelle 1.1: (continued)

Taste	**Funktion**	**Anmerkung**
Enter	**Ok**	Außerhalb des OSD-Menüs: kleine Info-Anzeige Jetzt & Nächste Sendung
Pfeil Auf	**Nach oben**	Außerhalb des OSD-Menüs: **Kanal hoch**
Pfeil Ab	**Nach unten**	Außerhalb des OSD-Menüs: **Kanal runter**
Pfeil links	**Nach links**	Außerhalb des OSD-Menüs Wechsel der Kanalgruppe
Pfeil rechts	**Nach rechts**	Außerhalb des OSD-Menüs Wechsel der Kanalgruppe
i	**Info**	Öffnet die Informationen zu einem Listenelement (z.B. Aufnahme, EPG-Eintrag, Timer)
p	**Power**	VDR herunterfahren, das Lifeguard-addon oder anstehende Timer können den Vorgang abbrechen bzw. verzögern.
F1	**Rot**	Die Belegung der Taste **"Rot"** wird jeweils im OSD eingeblendet.
F2	**Grün**	Die Belegung der Taste **"Grün"** wird jeweils im OSD eingeblendet.
F3	**Gelb**	Die Belegung der Taste **"Gelb"** wird jeweils im OSD eingeblendet.
F4	**Blau**	Die Belegung der Taste **"Blau"** wird jeweils im OSD eingeblendet.
F5	**Zurückspulen**	
F6	**Vorspulen**	
F10	**Mute**	Schaltet den Ton aus oder ein
F11	**Leiser**	
F12	**Lauter**	
Zifferntasten	**Ziffern 0-9**	

1.4.2.2 Konfiguration der Fernbedienung

Kein VDR wäre komplett ohne die Möglichkeit ihn bequem per Fernbedienung zu steuern. yaVDR 0.4 unterstützt ein breites Spektrum von Eingabegeräten, sowohl Infrarot- als auch Funk-Fernbedienungen. Falls die eigene Fernbedienung OOTB zufriedenstellend funktioniert kann man den folgenden Abschnitt überspringen. Wer keine Fernbedienungs-Empfänger Kombination hat, die OOTB funktioniert oder einfach verstehen will wie die Einbindung der Fernbedienungen in yaVDR funktioniert sollte einen Blick auf das Kapitel zur Fernbedienung und Eventlircd und zur remote.conf in dieser Dokumentation werfen.

Falls nicht alle Tasten wie erwartet funktionieren oder gar keine Tastendrücke beim VDR ankommen, konnte die Fernbedienung nicht automatisch durch yaVDR konfiguriert werden. Wie man dann weiter verfährt ist im Folgenden beschrieben:

Anfragen im VDR-Portal
Für eine Support-Anfrage im Forum ist es sinnvoll gleich ein paar Informationen über seinen Empfänger zu sammeln und die Ausgaben der hier in den einzelnen Schritten gezeigten Befehle gleich mit zu posten.



s2-liplianin-dkms
Außerdem sollte man sicherstellen, dass das Paket **s2-liplianin-dkms** nicht installiert ist, da u.U. **eventlircd** und **ir-keytable** ansonsten (Stand 08.11.11) nicht richtig funktionieren. Im Zweifelsfall sollte man das Paket **v4l-dvb-dkms** installieren, da es zum einen aktuellere Treiber und im Quellcode aktualisierte Keymaps einspielt und zum anderen **s2-liplianin-dkms** sicher verdrängt.

Im Folgenden ein kurzer Leitfaden, der dabei hilft notwendige Informationen über den verwendeten Empfänger zu sammeln. Dies erleichtert die Suche im Internet und hilft dabei Support-Anfragen im http://vdr-portal.de oder beim Hinzufügen der eigenen Konfiguration zur yaVDR-Konfigurationssammlung im Bug- & Feature-Tracker von yaVDR mit sinnvollen Informationen zu versehen.

Je nach Empfänger empfehlen sich unterschiedliche Herangehensweisen bei der Konfiguration. Um den Aufwand zu verringern folgt man am besten den nachfolgenden Links zum eigenen Empfänger.

Wie ist der Empfänger angeschlossen?

- Via USB
- Via TV-Karte
- Via Serielle Schnittstelle

Besonderheiten für bestimmte Fernbedienungen bzw. Empfänger

- X10-Empfänger
- Atric-Einschalter
- Igor-USB Empfänger

Andere Probleme

- Doppelte Tastendrücke, siehe FAQ
- Taste muss bei X10-Fernbedienung doppelt gedrückt werden: X10-Empfänger

1.4.2.3 USB-Empfänger und USB-DVB-Sticks

Anmerkung
Im folgenden sind alle Befehle mit den relevanten Ausgaben am Beispiel einer "Hama MCE Remote Control"beschrieben, die OOTB von yaVDR unterstützt wird, sich aber gut als Beispiel eignet.

Um zu sehen, welche USB-Geräte Linux momentan erkannt hat, kann man den Befehl lsusb benutzen. Er gibt Hinweise auf Hersteller und Typ der angeschlossenen Geräte.

lsusb

```
user@yavdr:~$ lsusb
                             ❶      ❷     ❸
Bus 002 Device 003: ID 05a4:9881 Ortek Technology, Inc.  ❹
Bus 002 Device 002: ID 80ee:0021
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

1. Hersteller-ID
2. Produkt-ID
3. Bezeichnung
4. Hama MCE Remote

Ruft man **dmesg** direkt nach dem Einstecken eines USB-Gerätes auf, sieht man, ob Linux für das Gerät Treiber geladen und Geräte angelegt hat.

dmesg

```
user@yavdr:~$ dmesg
[...]
[ 7178.064139] usb 2-2: new full speed USB device using ohci_hcd and address 3
[ 7178.535387] input: HID 05a4:9881 as /devices/pci0000:00/0000:00:06.0/usb2/2-2/2-2:1.0/ ↩
    input/input5
[ 7178.535645] generic-usb 0003:05A4:9881.0002: input,hidraw1: USB HID v1.10 Keyboard [HID ↩
    05a4:9881] on usb-0000:00:06.0-2/input0 ❶
[ 7178.574745] input: HID 05a4:9881 as /devices/pci0000:00/0000:00:06.0/usb2/2-2/2-2:1.1/ ↩
    input/input6
[ 7178.574885] generic-usb 0003:05A4:9881.0003: input,hidraw2: USB HID v1.10 Mouse [HID 05 ↩
    a4:9881] on usb-0000:00:06.0-2/input1 ❷
```

1, 2 Wie man sehen kann werden zwei Eingabegeräte (Fernbedienung + Mausfunktion) erkannt.

Für weitere Informationen dazu wie die Fernbedienung eingebunden wird schaut man sich die vom System erkannten Eingabegeräte an: cat /proc/bus/input/devices und folgt abhängig von der Ausgabe den dort verlinkten Abschitten der Dokumentation

1.4.2.4 Fernbedienungen von internen TV-Karten

Empfänger, die in TV-Karten eingebaut sind, werden normalerweise über rc-core eingebunden und mittels ir-keytable konfiguriert. Für einige Geräte gelten Ausnahmen von dieser Regel. Als Erstes sollte man also überprüfen, ob ein entsprechendes rc-core Gerät erkannt wurde:

Erkanntes rc-core Gerät

```
user@yavdr:~$ sudo ir-keytable
Found /sys/class/rc/rc0/ (/dev/input/event3) with:
        Driver ir-kbd-i2c, table rc-rc5-tv
        Supported protocols: RC-5
        Enabled protocols:
Repeat delay = 500 ms, repeat period = 33 ms
```

Die weitere Konfiguration ist im Abschnitt der Anleitung zu ir-keytable beschrieben.

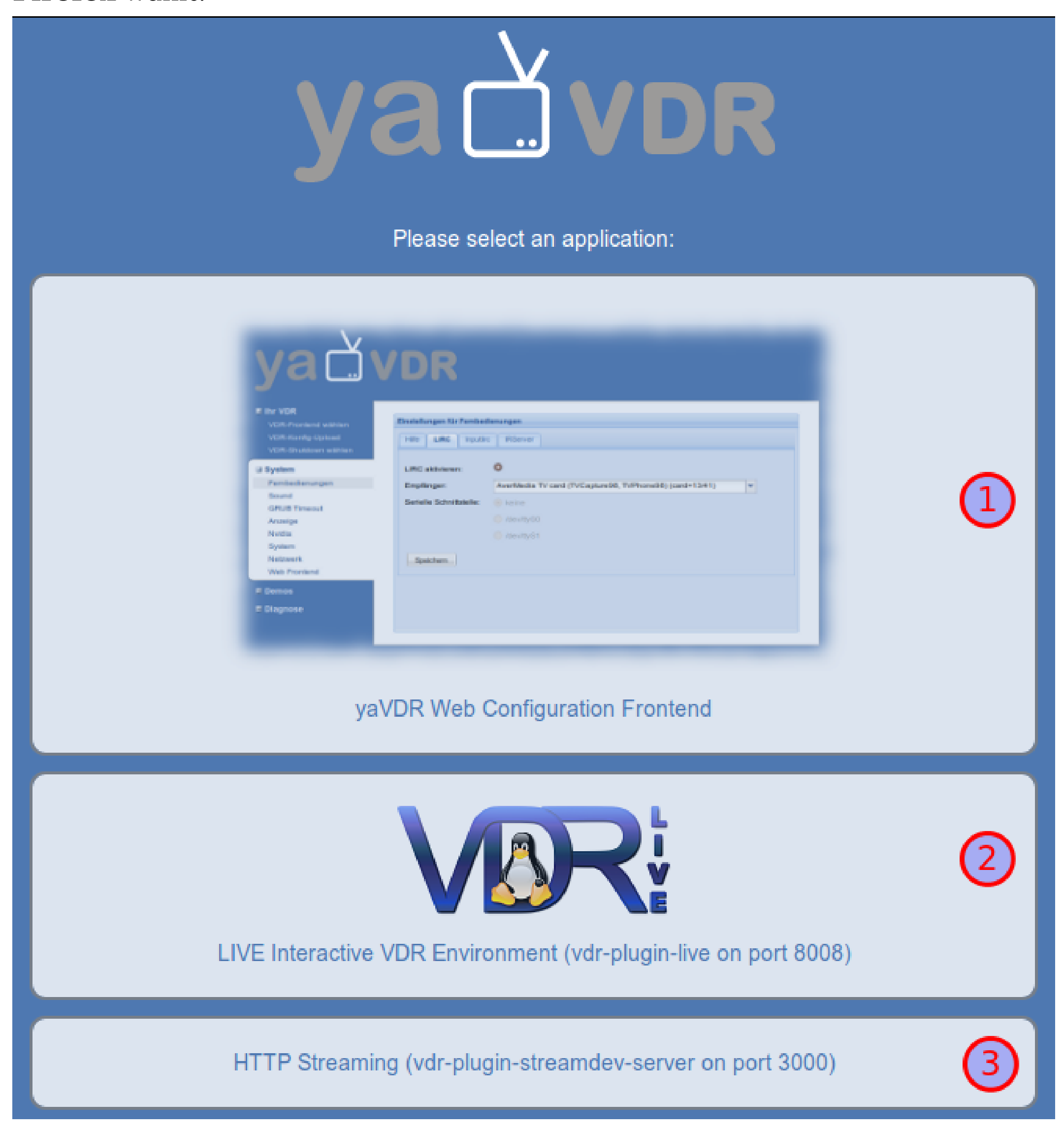
Falls kein derartiges Gerät erkannt wurde, taucht diese Ausgabe auf:

```
user@yavdr:~$ sudo ir-keytable
Couldn't find any node at /sys/class/rc/rc*.
```

Um herauszufinden, ob und wie die Fernbedienung eingebunden wird schaut man sich die vom System erkannten Eingabegeräte an: cat /proc/bus/input/devices

1.5 Das Webfrontend

Das Webfrontend ist via Browser unter der IP-Adresse oder dem Hostnamen des VDR aus dem Netzwerk erreichbar. Am yaVDR selbst ist es als Startseite im vorinstallieren Mozilla Firefox eingerichtet. Diesen kann man entweder über das VDR-OSD (Taste "M" auf der Tastatur, Menütaste auf der Fernbedienung: **Hauptmenü ⇒ "Applikationen" ⇒ "Web"**) aufrufen oder indem man die Maus mittig an den linken Bildschirmrand bewegt und aus dem nun erscheinenden Panel den Starter unter **http:// ⇒ Firefox** wählt.



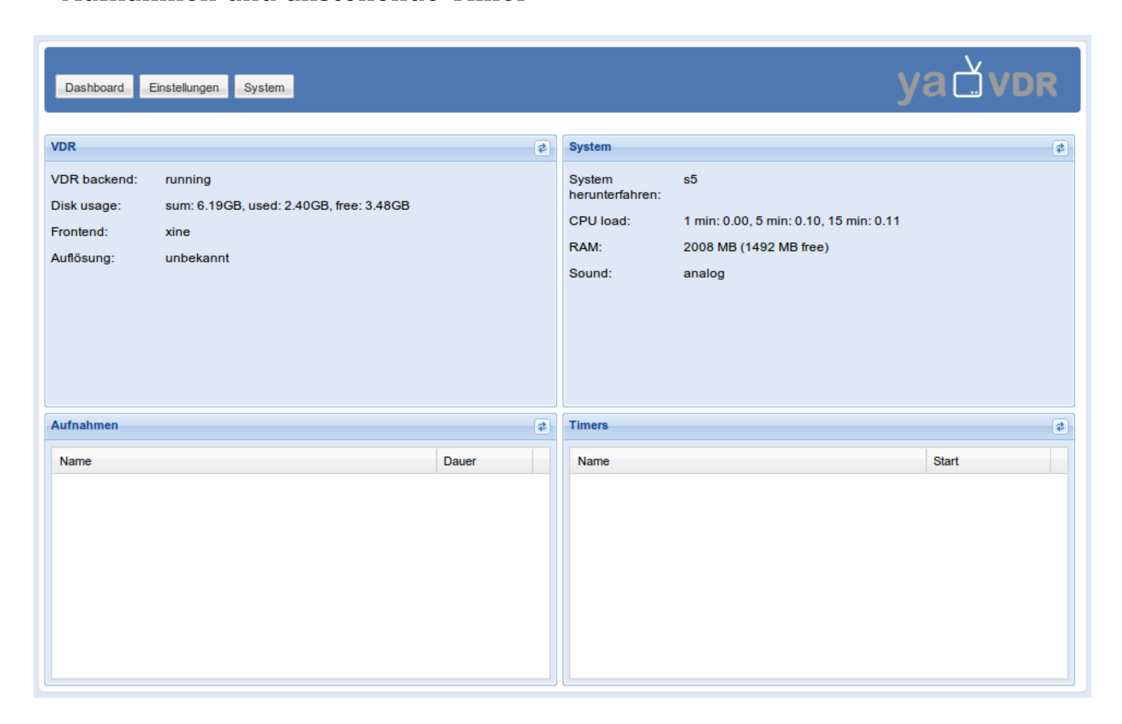
Die drei Einträge auf der Startseite verlinken auf:

1. Das yaVDR-Webfrontend zu Konfiguration
2. VDR LIVE!
3. Das HTTP-Interface des Streamdev-Servers

1.5.1 Dashboard

Das Webfrontend startet mit dem Dashboard, das eine Übersicht über das System gibt. Unter anderem wird angezeigt

- ob der VDR läuft
- freier und belegter Festplattenplatz
- das gewählte Frontend
- die Bildschirmauflösung
- Informationen zur Ausschaltmethode, CPU, RAM und Systemauslastung
- die gewählte Audioausgabeoption
- Aufnahmen und anstehende Timer



1.5.2 Überblick

Ein Großteil der Einstellungen kann bei yaVDR über das Webfrontend vorgenommen werden. Dazu gehören

- Wahl des VDR-Frontends
- Lifeguard-Addon
- EPG-Zeichenkorrektur
- Kanalverwaltung
- Fernbedienung(en)
- Tonausgabe
- Anzeigeneinstellungen
- Systemeinstellungen wie Sprache, Ausschalt- und Aufweckmethode
- Netzwerkeinstellungen zum Einbinden von NFS-Freigaben und dem Aufwecken anderer Rechner
- Paketverwaltung um Plugins für den VDR nachzuinstallieren
- Editor für Konfigurationsdateien

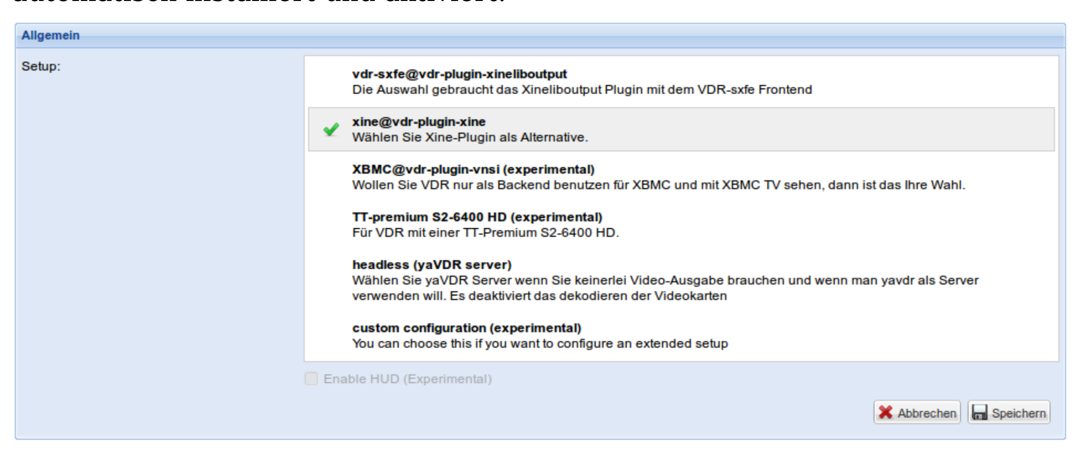
1.5.3 Einstellungen

Diese Seite verlinkt zu den Einstellungen, die über das Webfrontend vorgenommen werden können:

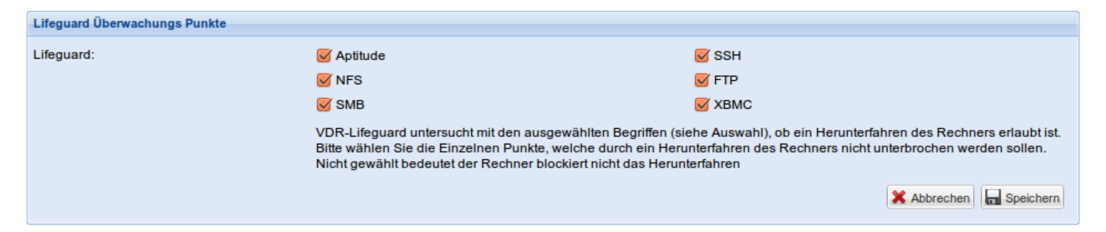
1.5.3.1 VDR: Allgemein

TODO:
Symbol hinzufügen

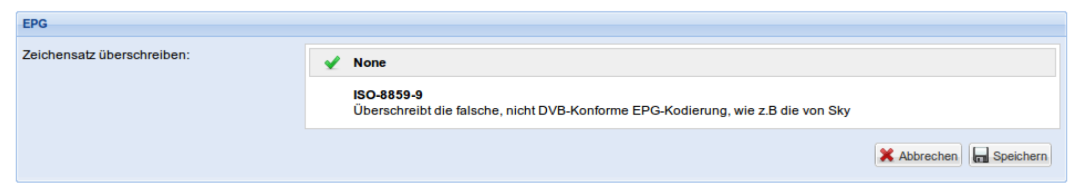
Hier kann das VDR-Frontend ausgewählt werden. bei yaVDR 0.4 ist xine vorausgewählt, wenn die Ausgabe über die Grafikkarte des PCs erfolgen soll. Ist eine TT S2-6400 (HD-FF), SD-FF oder eine Hauppauge PVR350 verbaut, werden die entsprechenden Plugins für diese Karten bei der Installation automatisch installiert und aktiviert.



Das Lifeguard-Addon erlaubt es ein automatisches Herunterfahren des VDRs zu verhindern, wenn noch aktive Verbindungen bzw. Prozesse der ausgewählten Dienste bestehen:



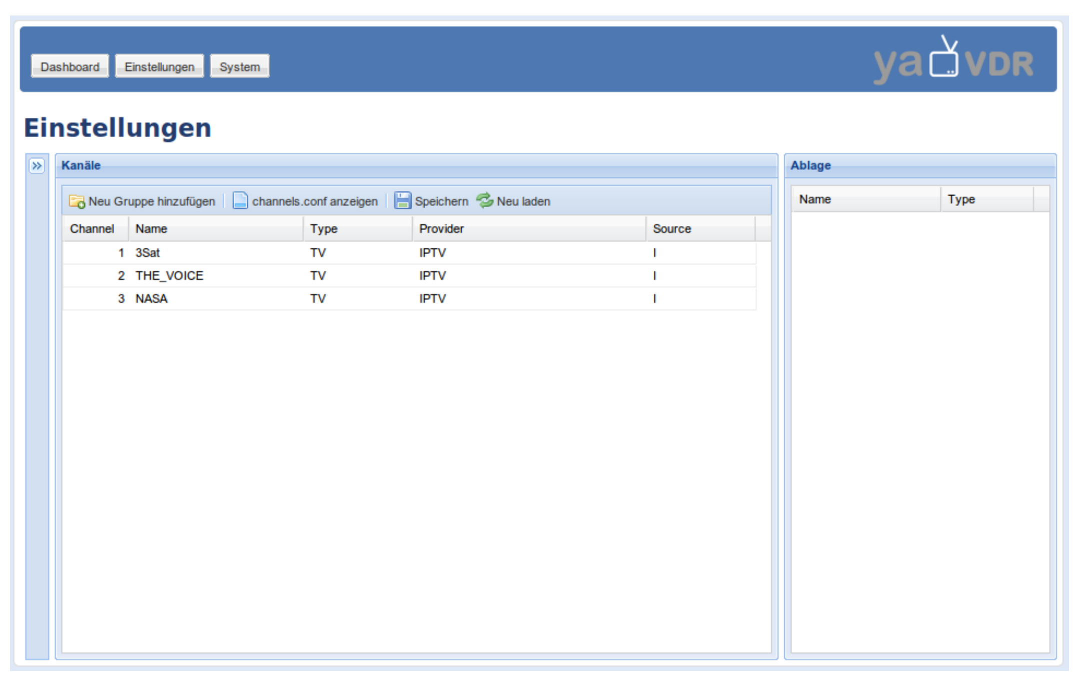
Für Sender, die den Elektronischen Programmführer nicht standardkonform encodiert senden, kann der VDR so eingestellt werden, dass er auch in diesen Fällen alle Sonderzeichen korrekt anzeigt.



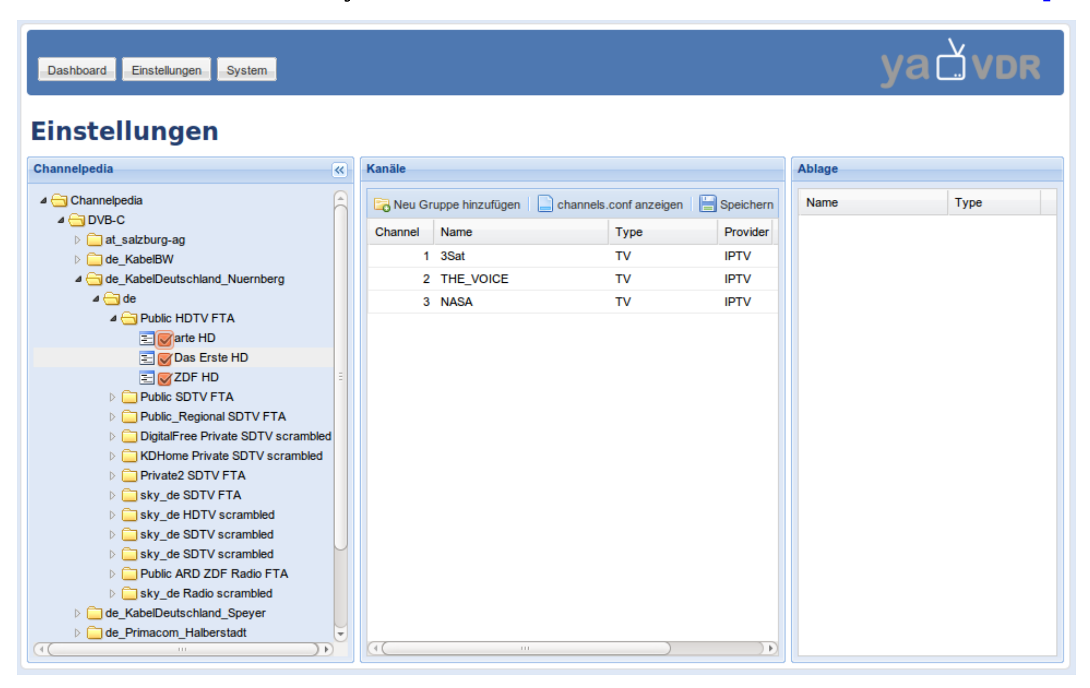
1.5.3.2 VDR: Kanäle

TODO:
Symbol hinzufügen

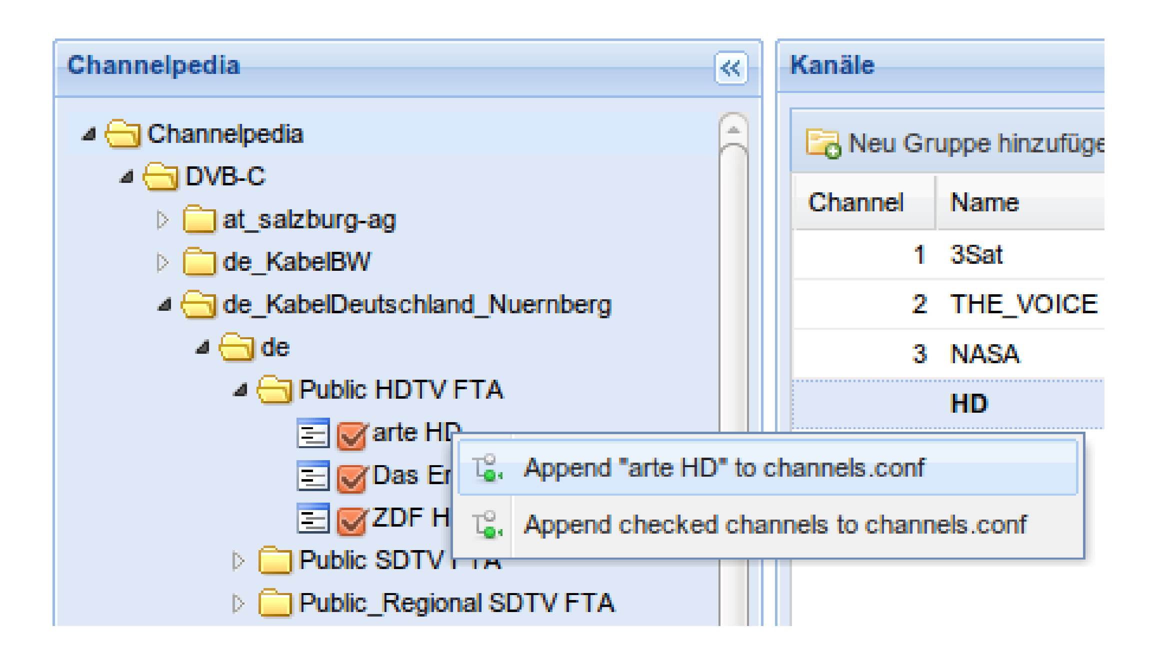
Hier können die Kanäle des VDR verwaltet werden. Vorhandene Kanäle lassen sich in Gruppen einteilen, per Drag and Drop neu anordnen oder über das Kontextmenü, das per Rechtsklick erreichbar ist, bearbeiten. Mehrere Kanäle lassen sich bei gedrückter Umschalttaste gleichzeitig auswählen. Die Ablage rechts erlaubt es Kanäle zwischenzulagern, z.B. um sie ohne ständiges Hin- und Herscrollen anordnen zu können.



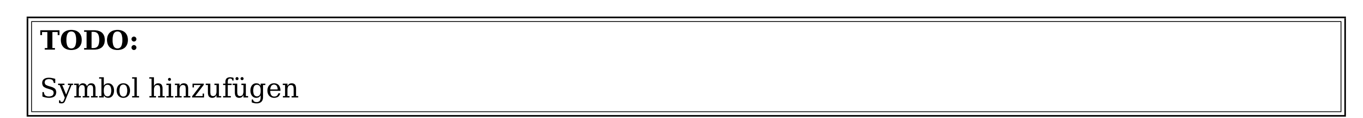
Klickt man auf das Symbol links neben "Kanäle", öffnet sich die Channelpedia:



Dort hinterlegte Sender können ohne Kanalsuchlauf direkt in die Kanalliste übernommen werden. Dazu ruft man mittels Rechtsklick das Kontextmenü eines Kanals oder einer Kanalgruppe auf und übernimmt wahlweise das gerade markierte Objekt oder alle per Häkchen ausgewählten Objekte in die Kanalliste.



1.5.3.3 Hardware: Fernbedienung





yaVDR nutzt Eventlircd, ein Systemdienst, der viele Fernbedienungsempfänger automatisch erkennt und für die mit ihnen ausgelieferte Fernbedienung konfiguriert. Diese Fernbedienungen funktionieren dann ohne weiter Konfiguration, in den Einstellungen des Webfrontends zur Fernbedienung muss nichts angepasst werden.

Für spezielle Empfänger (die z.B. Atric-Einschalter und andere, die über die Serielle Schnittstelle angeschlossen sind, in Ausnahmefällen auch für X10-Empfänger) kann Lirc aktiviert werden. Für serielle Empfänger muss außerdem angegeben werden, welche serielle Schnittstelle benutzt werden soll. Auch hier sind für einige Empfänger die jeweiligen Tasten der Fernbedienung vorkonfiguriert. Ist dies nicht der Fall oder wird eine andere Fernbedienung verwendet, muss die */etc/lirc/lircd.conf* angepasst werden. Dabei ist zu beachten, dass die Tastennamen **namespacekonform** sind und sich an der Benennung in der */etc/vdr/remote.conf* orientieren. So ist sichergestellt, dass die Fernbedienungstasten sowohl mit dem VDR als auch unter XBMC funktionieren. Eingaben von Lircd werden unter Nutzung von uinput (Userland Input) an Eventlircd weitergereicht und werden so wie alle übrigen Tastendrücke von Eventlircd-Eingabegeräten an den VDR weitergegeben.



Igor-USB
Für Igor-USB Empfänger darf Lirc nicht über das Webfrontend aktiviert werden. Zur weiteren Konfiguration siehe Igor-USB Empfänger

1.5.3.4 Hardware: Audio

TODO:

Symbol hinzufügen

Diese Seite erlaubt es einzustellen wie der Ton durch den VDR und XBMC ausgegeben werden soll. Für komplexere Konstellationen und mehreren Soundkarten im Rechner muss die Konfiguration von alsa ggf. von Hand angepasst werden.

1.5.3.5 Hardware: Anzeigeneinstellungen

TODO:

Symbol hinzufügen

Hier kann die Auflösung und Nutzung der angeschlossenen Bildschirme eingestellt werden. Außerdem können hier die Deinterlacing-Methoden für HD- und SD-Material konfiguriert werden.

GraphTFT

Das GraphTFT-Plugin erlaubt es Statusinformationen und das Menü des VDR auf einem zweiten angeschlossenen Bildschirm auszugeben.

GraphTFT Aktivieren
Momentan ist die Konfiguration eines zweiten Bildschirms in yaVDR noch etwas tricky. Hier ein Weg der am ehesten zum Erfolg führt:

- Monitore neu erkennen lassen ("Rescan Displays")
- Neustart, damit die Reihenfolge der Anzeigen sicher stimmt
- Dualscreen einrichten
- Graphtft mit anhaken und dabei Auflösung und Wiederholrate des GraphTFT-Bildschirms von Hand festlegen
- Falls die Reihenfolge noch nicht stimmt primären und sekundären Monitor festlegen

1.5.3.6 System: System

TODO:

Symbol hinzufügen

Auf dieser Seite kann die Systemsprache gewählt werden und eingestellt werden wie der VDR herunterfährt und aufwacht. Außerdem lässt sich der Timeout für die Anzeige des GRUB-Menü einstellen.

1.5.3.7 System: Netzwerk

> **TODO:**
>
> Symbol hinzufügen, Bilder hinzufügen, Beispiel hinzufügen

Hier können NFS-Freigaben aus dem Netzwerk und die MAC-Adressen von Rechnern, die später über das OSD per WOL [9] geweckt werden sollen, eingetragen werden.

1.5.3.8 VDR: Konfiguration bearbeiten

> **TODO:**
>
> Symbol hinzufügen

Hier können Konfigurationsdateien direkt im Browser bearbeitet werden. Aktuell können folgende Konfigurations-Dateien über das Webfrontend bearbeitet werden:

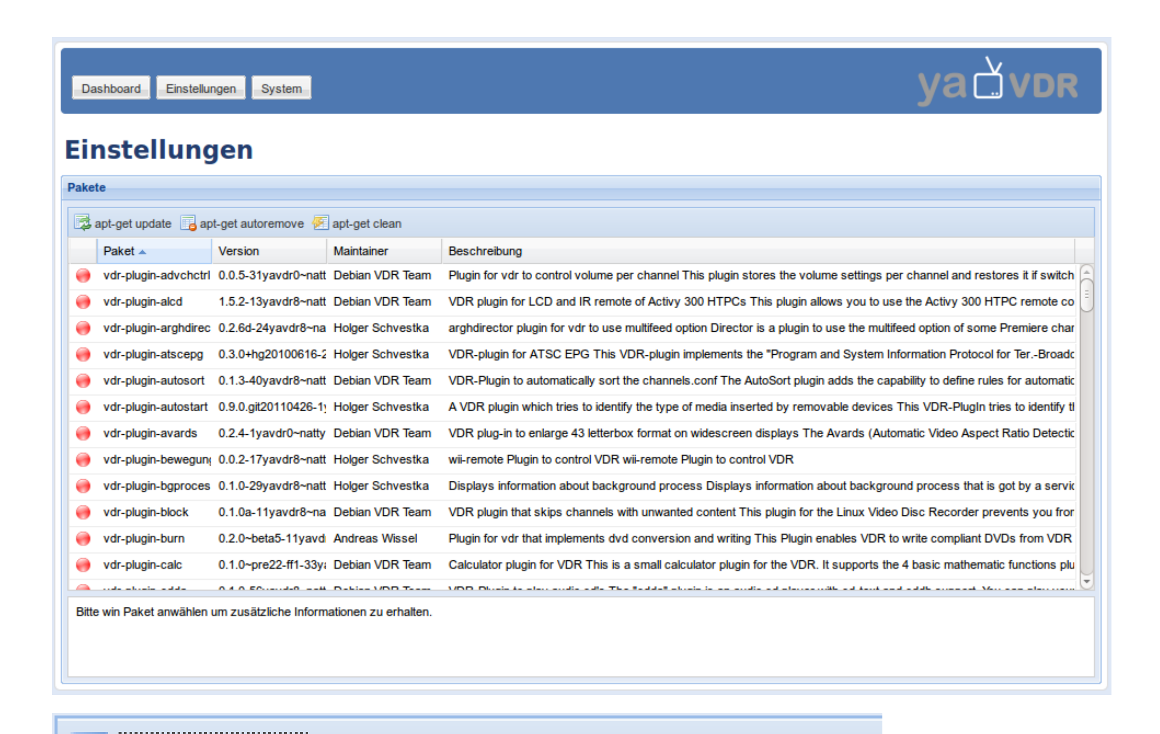
- */etc/lirc/lircd.conf*
- */etc/vdr/diseq.conf*
- */var/lib/vdr/remote.conf*
- */var/lib/vdr/channels.conf*
- */var/lib/vdr/.xbmc/userdata/Lircmap.xml*

1.5.3.9 System: Pakete

> **TODO:**
>
> Symbol hinzufügen

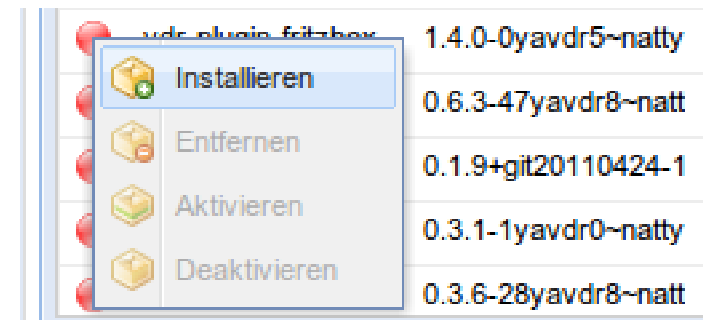
Über das Webfrontend können Plugins, Addons und andere VDR-bezogene Pakete verwaltet werden.

[9] Wake On LAN



Über die Buttons über der Plugin-Liste können die Paketquellen neu eingelesen ("apt-get update"), nicht mehr benötigte als Abhängigkeit installierte Pakete deinstalliert ("apt-get autoremove") und lokale Kopien heruntergeladener Pakete gelöscht werden ("apt-get clean").

Über das mit einem Rechtsklick auf ein Paket in der Liste erreichbare Menü kann das jeweilige Paktet installiert, entfernt oder (de)aktiviert werden.



1.5.4 System

TODO: Unterpunkte ausführen

1.5.4.1 Befehle

- Switch temporarily to second screen
- Reboot computer
- Kill XBMC

- Set XBMC defaults
- Restart VDR

1.5.4.2 Diagnosewerkzeuge

- System information
- System log files
- XBMC crash log
- LIRC configuration
- VDR configuration
- Xorg server
- Sound (ALSA)
- Package information
- yaVDR database

Kapitel 2

Systembeschreibung

2.1 Verzeichnisstruktur

Für **yaVDR** werden verschiedene Dienste und Methoden kombiniert um eine Verzeichnisstruktur zu erzeugen, die es erlaubt lokale und von anderen Computern eingebundene Freigaben zusammenzuführen und gemeinsam im Dateisystem abzubilden.

Für lokal abgelegte Musik, Filme und Bilder existieren bereits Verzeichnisse unter */srv/*, die per NFS und SAMBA im Netzwerk freigegeben sind, die Aufnahmen des VDR liegen unter */srv/vdr/video.00* und werden als **recordings** via Samba exportiert (siehe Tabelle)

Tabelle 2.1: Verzeichnisse für lokale Medien in yaVDR

Kategorie	Pfad	NFS	SAMBA
Bilder	***/srv/picture***	***/srv/picture***	**picture**
Musik	***/srv/audio***	***/srv/audio***	**audio**
Videos	***/srv/video***	***/srv/video***	**video**
Aufnahmen	***/srv/vdr/video.00*** [1]	***/srv/share/vdr***	**recordings**

TODO:

Samba in fstab usw.

2.1.1 Avahi-Mounter

Der Avahi-Mounter ist ein Python-Skript, das nach Avahi-Announcements für Freigaben im Netzwerk sucht und diese - falls eine gültige Service-Konfiguration existiert - automatisch zusammen mit den lokalen Medien unter */media/<Kategorie>/<Ort der Dateien>* einbindet. Für Rechner mit yaVDR 0.4 geschieht das automatisch für alle lokalen Standardverzeichnisse.

Möchte man also sowohl lokale als auch über das Netzwerk eingebundene Dateien für VDR-Plugins oder XBMC nutzen, empfiehlt es sich als Standard-Verzeichnis */media/<Kategorie>/* zu wählen.

[1] Bei mehren video.xx-Verzeichnissen entsprechend zusätzlich /srv/vdr/video.01, /srv/vdr/video.02 usw.

Beispiel für die Dateistruktur unter */media/* auf einem yaVDR 0.4 System, das Verzeichnisse von einem weiteren Rechner mit yaVDR 0.4 im Netzwerk mit dem Namen "VDR" automatisch eingebunden hat. Wie man sieht sind die lokal abgelegten Medien jeweils über einen Ordner "local" verlinkt:

Ausgabe von tree

```
/media├──
 Bilder│
   ├── local -> /srv/picture│
   └── VDR├──
 cdrom├──
 Musik│
   ├── local -> /srv/audio│
   └── VDR├──
 Record├──
 Sonstige└──
 Video├──
     local -> /srv/video└──
     VDR

/srv/vdr/video.00/└──
 VDR
```

2.1.2 NFS Freigaben mounten

Unter yaVDR gibt es mehrere Möglichkeiten NFS-Freigaben zu mounten. Wie man Freigaben auf einem Server announciert, die per Avahi-Mounter eingebunden werden kann man in diesem Artikel auf yavdr.org nachlesen. Verzeichnisse, die Aufnahmen enthalten können über das Webfrontend unter dem Punkt Netzwerk eingebunden werden.

2.2 PPAs für yaVDR

yaVDR nutzt die **stabilen PPAs** [2] des yaVDR-Teams. Wirft man einen Blick in die Datei */etc/apt/sources.list.d/yavdr.list*, kann man sehen, dass 4 PPAs und ihre Quellen eingebunden sind:

/etc/apt/sources.list.d/yavdr.list

```
# yaVDR-PPAs für yaVDR 0.4
deb http://ppa.launchpad.net/yavdr/main/ubuntu natty main            ❶
deb http://ppa.launchpad.net/yavdr/stable-vdr/ubuntu natty main      ❷
deb http://ppa.launchpad.net/yavdr/stable-xbmc/ubuntu natty main     ❸
deb http://ppa.launchpad.net/yavdr/stable-yavdr/ubuntu natty main    ❹

# Quellen für die Pakete in yaVDR 0.4
deb-src http://ppa.launchpad.net/yavdr/main/ubuntu natty main
deb-src http://ppa.launchpad.net/yavdr/stable-vdr/ubuntu natty main
deb-src http://ppa.launchpad.net/yavdr/stable-xbmc/ubuntu natty main
deb-src http://ppa.launchpad.net/yavdr/stable-yavdr/ubuntu natty main
```

❶ In **main** liegen die DVB- und Grafiktreiber, angepasste Software-Pakete, VDR-Addons usw.

❷ **stable-vdr** beinhaltet den VDR und seine Plugins

[2] **PPA** = **P**ersonal **P**ackage **A**rchive, siehe http://wiki.ubuntuusers.de/Launchpad/PPA

3. **stable-xbmc** ist das PPA für XBMC
4. **stable-yavdr** enthält alle yaVDR-spezifischen Erweiterungen, wie das Webfrontend, die Upstart-Skripte, das Template-System usw.



Verwendung der yaVDR PPAs ohne yaVDR
Es ist prinzipiell möglich die PPAs **main**, **unstable-/testing-/stable-vdr** und **unstable-/testing-/stable-xbmc** auf einem Ubuntu-System zu verwenden, das nicht mit Hilfe der yaVDR-Installations-CD installiert wurde. Wichtig dabei ist, dass das main-PPA **immer** mit eingebunden wird, da viele Pakete aus den anderen PPAs dieses benötigen um Abhängigkeiten aufzulösen. Für dabei auftretende Probleme muss sich der Nutzer dann aber selbstständig um eine Lösung bemühen, solche Posts gehören **nicht** in das yaVDR-Unterforum, sondern in das Unterforum zu Debian und seinen Derivaten.
Das PPA **unstable-/testing-/stable-yavdr** lässt sich nicht ohne weiteres verwenden und kann die Funktionsfähigkeit eines Systems stören, da die dort verwendeten Upstartskripte und andere yaVDR-spezifische Anpassungen z.B. mit einem "normalen" Ubuntu-Desktop-System kollidieren. Im VDR-Portal wurde beschrieben, dass man mit einigen Tricks ausgehend von einer Ubuntu-Minimalinstallation zu einem **beinahe-yaVDR** System kommen kann (z.B. für 32-Bit Rechner). Diese Anleitung funktionierte offenbar zum Zeitpunkt ihrer Erstellung, im Zuge der Weiterentwicklung von yaVDR kann aber nicht ausgeschlossen werden, dass durch künftige Änderungen an der Installationsmethode von yaVDR dieser Workaround nicht mehr funktionieren könnte. **Das Beheben von Fehlern bei auf diese Weise installierten Systemen hat für das yaVDR-Team keine Priorität.**

2.3 Fernbedienung

Kein VDR wäre komplett ohne die Möglichkeit ihn bequem per Fernbedienung zu steuern. yaVDR 0.4 unterstützt ein breites Spektrum von Eingabegeräten, sowohl Infrarot- als auch Funk-Fernbedienungen. Ob deine Fernbedienung unterstützt wird, kannst du

hoffentlich bald hier lesen - Liste im yaVDR-Wiki sinnvoll?

Das yaVDR-Projekt lebt von deiner Mithilfe.
Bitte schicke den Entwicklern die Konfigurationsdateien für deine noch nicht OOTB [a] funktionsfähige Fernbedienung, damit die Hardware-Unterstützung für die kommenden Versionen verbessert werden kann.

[a] Out of the box, d.h. direkt nach der Installation von yaVDR funktionierend

2.3.1 Eventlircd

yaVDR 0.4 benutzt standardmäßig eventlircd, ein Dienst der dabei hilft die unterschiedlichen Treiber und Eingabegeräte unter einen Hut zu bringen. Es dient sozusagen als Filter, der die Tastendrücke aller konfigurierten Fernbedienungen auswertet, auf Wunsch die Tastencodes vereinheitlicht und an den VDR weiterreicht.

2.3.1.1 Die Theorie [3]

[3] http://www.vdr-portal.de/board60-linux/board14-betriebssystem/board96-yavdr/p1009867-yavdr-0-4-eventlircd-und-fernbedienungen-die-grundlagen/#post1009867

Eventlircd übersetzt Tastendrücke von Eingabegeräten [4] in Lirc-Tastendrücke. Im Prinzip ähnelt die Funktion dem remote-plugin mit dem Unterschied, dass nur Tasten ausgewertet werden, denen über eine Keymap [5] eine Funktion zugewiesen wurde.

Solche Eingabegeräte sind zum Beispiel normale Tastaturen oder auch Fernbedienungen die über ihre Treiber als Tastaturen eingebunden werden. Fernbedienungen welche an DVB Karten angeschlossen sind, melden sich teilweise ebenfalls als Tastaturen, außerdem existiert eine neue Schnittstelle für Fernbedienungen namens rc-core, auf welche Jarod Wilson [6] schon einige der bestehenden Lirc Treiber und auch einige Konfigurationen der mit DVB Geräten ausgelieferten Fernbedienungen portiert hat. Auch diese Schnittstelle - auch wenn sie kein normaler Tastaturtreiber ist - stellt ihre Geräte als Eingabegeräte zur Verfügung.

Eingaben von Zeigegeräten (wie Maus, Touchpad, Grafiktablett), die ebenfalls über die Input-Schnittstelle arbeiten, versucht eventlircd unverändert durchzuleiten.

Eventlircd erkennt Eingabegeräte über udev, auch wenn sie erst nach dem Start des Dämon eingebunden oder angeschlossen werden. Hier liegt einer der Vorteile dieser Lösung, da eventlircd ab seinem Start einen Lirc Socket für den VDR bereitstellt, und nach und nach die ihm zugeordneten Geräte als Eingabegeräte einbinden kann.

Um welche Geräte es sich kümmern soll, erkennt eventlircd anhand von Attributen, welche man den Geräten bei der Initialisierung über udev-Regeln mitgeben kann. Für manche Geräte funktioniert das schon in der Standardkonfiguration (rc-core, Eingabegeräte die das yaVDR-Team selber besitzt, Konfigurationen die an das yaVDR-Team weitergegeben wurden).

Lircd [7] kann man über den Switch [8] --uinput [9] anweisen seine Tastendrücke an ein Gerät unter */etc/input/* zu übergeben. Damit kann dann lircd parallel zu anderen Eingabegeräten gestartet werden und es ist so theoretisch auch möglich mehrere Instanzen von lircd für verschiedene Geräte mit unterschiedlicher Konfiguration gleichzeitig starten. Für USB-Lirc Empfänger wird über eine udev-Regel in */lib/udev/rules.d/98-lircd.rules* und dem lircd_helper eine Instanz des Lirc-Daemon gestartet und die in der udev-Regel angegebenen Treiber und lircd.conf benutzt. Für diese Empfänger ist keine Konfiguration über das Webfrontend erforderlich. Weitere noch nicht OOTB unterstützte Lirc-Empfänger sollten entsprechend über ergänzende udev-Regeln nachgepflegt werden.



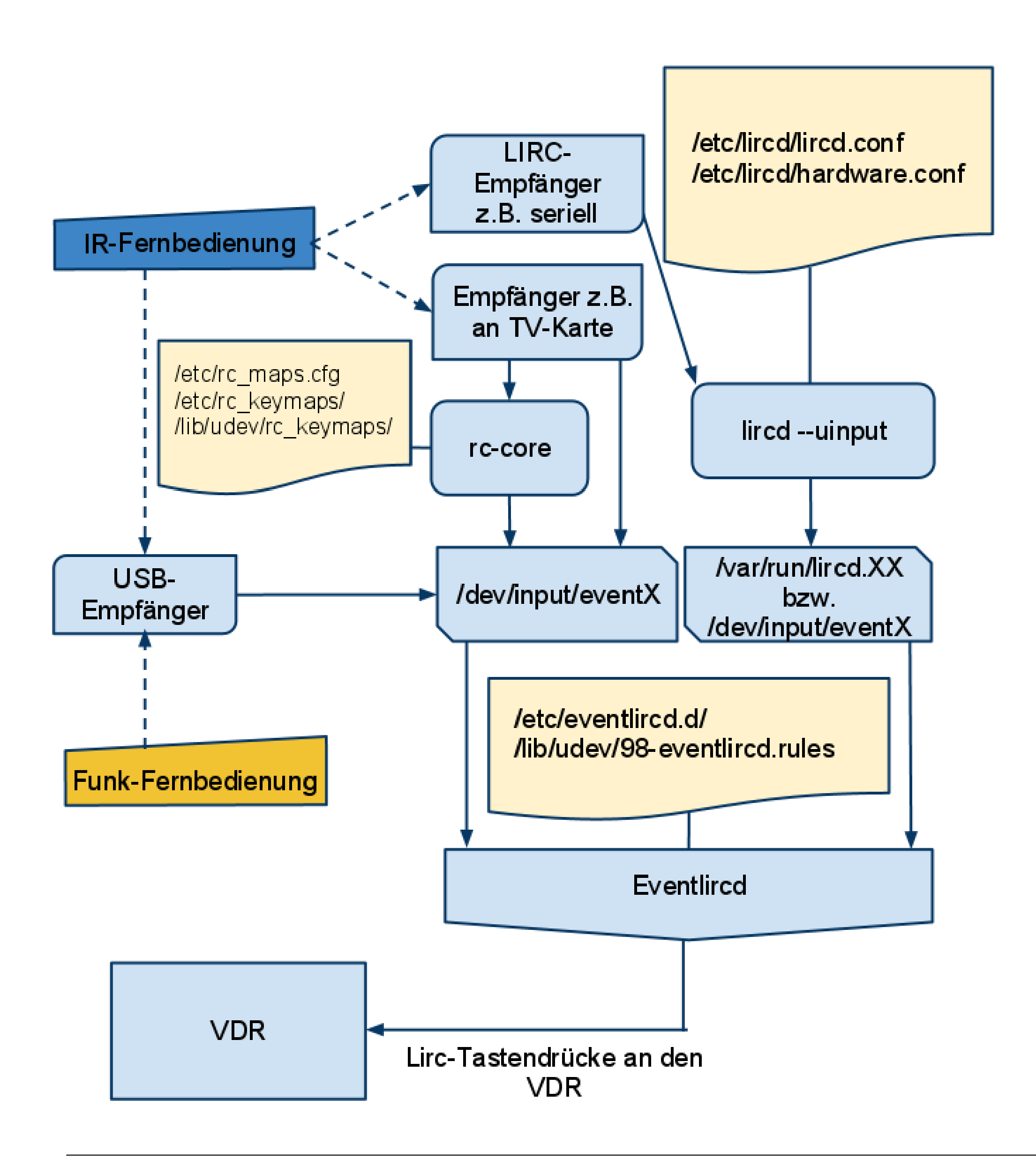
Bug bei Verwendung von --uinput
Leider scheinen aktuelle Lirc-Versionen unter einem Bug bei der Übergabe von Tastendrücken bei Verwendung von **--uinput** zu leiden - es kommen dann doppelte Tastendrücke am Eventgerät an (die Tastendrücke an eigentlichen Lircd-Socket hingegen sind in Ordnung). Ein Workaround ist hier für über das WFE konfigurierte und hier für per udev-Regel eingebundene USB-Lirc-Empfänger im Forum beschrieben und wird voraussichtlich seinen Weg in die nächste yaVDR-Version finden.

Die Ausgabe erfolgt nach Verarbeitung durch Eventlircd für die Anwendungen dann so, als würde sie von einem einzigen Gerät stammen.

Da manche Geräte eindeutig erkannt werden können (zum Beispiel USB Geräte), kann man diese über udev-Regeln mit der für das jeweilige Geräte richtigen Konfiguration automatisch einbinden. Die Konfigurationen für all diese Geräte kann man alle gleichzeitig auf dem System haben, da durch udev nur die Konfiguration für angeschlossene Geräte aktiviert wird.

Eventlircd kann geräte- oder treiberabhängig eine Übersetzungstabelle mitgegeben werden, um die Tastenbelegung anzupassen. (sogenannte evmaps)

[4] bzw. "Eventgeräten", also Eingabegeräte, für die durch den Treiber ein Eintrag unter */dev/input/* erstellt wird
[5] Konfigurationsdatei, die für die Übersetzung des Tastendrucks auf der Fernbedienung in eine Taste am Computer herangezogen wird
[6] Neuer Maintainer des Lirc Projektes
[7] Der LIRC-Daemon
[8] Das Startargument
[9] User Level Input Subsystem - eine Schnittstelle, um Eingaben aus dem Userland an den Kernel zu übermitteln



Alle Wege führen zu Eventlircd
Je nachdem welchen Anschluss und welche Treiber die eigenen Empfänger bzw. TV-Karten verwenden kann der Weg eines Tastendrucks also sehr unterschiedlich sein, bis er von eventlircd ausgewertet wird. Dieses Hintergrundwissen ist dann von Vorteil, wenn die Tastenbelegung die Eventlircd ausgibt nicht zur **remote.conf** von yaVDR passt.

Die Vorteile sind also:

- frühe Verfügbarkeit des Lircsocket (und damit schnellerer Start des VDR)
- Vorkonfiguration für verschiedenste Eingabegeräte möglich
- Tastatur- , rc-core- und Lirc-Geräte können angesprochen werden.
- Die Konfiguration durch den Nutzer entfällt, wenn die Fernbedieung und der Empfänger den yaVDR-Entwicklern bereits bekannt sind und schon unterstützt werden.

Wie bei jedem anderen Konzept gibt es natürlich auch "Nachteile":

- Um die Vorteile der Lösung mit Eventlircd zu nutzen, braucht man eine Standardbelegung auf die man die Tasten normalisieren kann, dadurch kommt dann die Anlernfunktion des VDR nicht mehr zum tragen. Wenn das Gerät noch nicht unterstützt wird, bedeuteutet dies unter Umständen ein etwas mühsames manuelles Anpassen bis die Tasten der eigenen Fernbedienung auf die yaVDR-Team mitgelieferte Belegung für Lirc in der */var/lib/vdr/remote.conf* passt.
- Um in den Genuß zu kommen Lirc wie oben beschrieben nutzen zu können, müssen die Tasten in der lircd.conf entsprechend der Regeln der Inputgeräte benannt werden. Bei Lirc heisst das Namespace-konform benannt sein. Sinnvoll wäre hier natürlich eine Benennung entsprechend unserer Standardbelegung.

Bei der yaVDR-Standardbelegung haben sich die Entwickler an der Dokumentation von linux-media (Tabelle am Ende der Seite) orientiert und benutzen diese als Referenz, eine Zuordnung von VDR-Taste und Tastenname kann man sich hier ansehen: neue IR-Integration für yaVDR 0.4

2.3.1.2 remote.conf



Keine Änderungen für die Fernbedienung an der remote.conf!
yaVDR ist auf eine standardisierte remote.conf angewiesen, damit die Fernbedienungen richtig funktionieren. Das hat zur Folge, dass die Tastennamen in den Lirc-Konfigurationsdateien, den Keytables von ir-core und den von Eventlircd direkt angesprochenen Geräten **Namespace-konform** zur remote.conf sein müssen. Die Anpassung des Mappings ist in diesem Kapitel beschrieben.

Die remote.conf liegt unter /var/lib/vdr/remote.conf und ist eine getemplatete Datei. Sie sollte möglichst nicht verändert werden (Ausnahme ist die Konfiguration der Tastenbelegung für Tastaturen), und gibt vor wie die von den Fernbedienungen gelieferten Tastennamen lauten sollten. Die Benennung ist gleichzeitig auf die XBMC-Konfiguration abgestimmt, so dass auch XBMC ohne weitere Anpassung seiner Keymaps bedienbar sein sollte.

Die Zuordnung von Funktion im VDR und Tastennamen sieht bei yaVDR 0.4 so aus:

Tabelle 2.2: Namespace-konforme Tastennamen in yaVDR

remote.conf	Tastenname durch Eventlircd	Beschreibung
LIRC.Up	**KEY_UP**	*Nach oben*
LIRC.Down	**KEY_DOWN**	*Nach unten*
LIRC.Menu	**KEY_MENU**	*Menü*
LIRC.Ok	**KEY_OK**	*OK/Eingabe*
LIRC.Back	**KEY_ESC**	*Zurück*
LIRC.Left	**KEY_LEFT**	*Nach links*

Tabelle 2.2: (continued)

remote.conf	Tastenname durch Eventlircd	Beschreibung
LIRC.Right	**KEY_RIGHT**	*Nach rechts*
LIRC.Red	**KEY_RED**	*Rot*
LIRC.Green	**KEY_GREEN**	*Grün*
LIRC.Yellow	**KEY_YELLOW**	*Gelb*
LIRC.Blue	**KEY_BLUE**	*Blau*
LIRC.0	**KEY_0**	*Ziffer 0*
LIRC.1	**KEY_1**	*Ziffer 1*
LIRC.2	**KEY_2**	*Ziffer 2*
LIRC.3	**KEY_3**	*Ziffer 3*
LIRC.4	**KEY_4**	*Ziffer 4*
LIRC.5	**KEY_5**	*Ziffer 5*
LIRC.6	**KEY_6**	*Ziffer 6*
LIRC.7	**KEY_7**	*Ziffer 7*
LIRC.8	**KEY_8**	*Ziffer 8*
LIRC.9	**KEY_9**	*Ziffer 9*
LIRC.Info	**KEY_INFO**	*Info zum aktuellen Objekt*
LIRC.Play	**KEY_PLAY**	*Play*
LIRC.Pause	**KEY_PAUSE**	*Pause*
LIRC.Stop	**KEY_STOP**	*Stop*
LIRC.Record	**KEY_RECORD**	*Aufnehmen*
LIRC.FastFwd	**KEY_FASTFORWARD**	*Vorspulen*
LIRC.FastRew	**KEY_REWIND**	*Zurückspulen*
LIRC.Next	**KEY_NEXT**	*Vorwärts springen*
LIRC.Prev	**KEY_BACK**	*Rückwärts springen*
LIRC.Power	**KEY_POWER2**	*Ausschalten*
LIRC.Channel+	**KEY_CHANNELUP**	*Kanal hoch*
LIRC.Channel-	**KEY_CHANNELDOWN**	*Kanal runter*
LIRC.PrevChannel	**KEY_PREVIOUS**	*zurück zum zuvor gewählten Kanal*
LIRC.Volume+	**KEY_VOLUMEUP**	*Lautstärke erhöhen*
LIRC.Volume-	**KEY_VOLUMEDOWN**	*Lautstärke verringern*
LIRC.Mute	**KEY_MUTE**	*Stummschalten*
LIRC.Subtitles	**KEY_SUBTITLE**	*Untertitel einblenden*
LIRC.Schedule	**KEY_EPG**	*Programmführer*
LIRC.Channels	**KEY_CHANNEL**	*Kanäle*
LIRC.Commands	**KEY_FAVORITES**	*Befehlsauswahl*
LIRC.Audio	**KEY_MODE**	*Tonspur wählen*
LIRC.Timers	**KEY_TIME**	*gesetzte Timer*
LIRC.Recordings	**KEY_PVR**	*Aufnahmen*
LIRC.Setup	**KEY_SETUP**	*Setup-Menü*
LIRC.User0	**KEY_TEXT**	*aktiviert Teletext bei installiertem teletext-plugin*
LIRC.User1	**KEY_PROG1**	*User-Taste, z.B. für Keymakros*
LIRC.User2	**KEY_PROG2**	*User-Taste, z.B. für Keymakros*
LIRC.User3	**KEY_PROG3**	*User-Taste, z.B. für Keymakros*
LIRC.User4	**KEY_PROG4**	*User-Taste, z.B. für Keymakros*
LIRC.User5	**KEY_AUDIO**	*User-Taste, z.B. für Keymakros*
LIRC.User6	**KEY_VIDEO**	*User-Taste, z.B. für Keymakros*
LIRC.User7	**KEY_IMAGES**	*User-Taste, z.B. für Keymakros*
LIRC.User8	**KEY_FN**	*User-Taste, z.B. für Keymakros*
LIRC.User9	**KEY_SCREEN**	*User-Taste, z.B. für Keymakros*

2.3.1.3 evmaps



Wann sind evmaps sinnvoll?
evmaps sollten nur für Geräte angelegt werden, die keine andere Möglichkeit bieten die Tastennamen zu verändern. Für Lirc- und rc-core-Geräte sollte man vorzugsweise eine entsprechene Konfigurationsdatei (lircd.conf bzw. Keymap) zur korrekten Belegung der Tasten erstellen.

Alle Tastendrücke, die an Eventlirc weitergereicht werden, können durch Übersetzungstabellen, die sogenannten evmaps [10] noch verändert werden, bevor sie an den VDR übergeben werden. Sie befinden sich unter */etc/eventlircd.d/* und sind entweder nach dem Gerätenamen, dem Treiber (z.B. *ircore.evmap* für Geräte, die mit **rc-core** als Treiber arbeiten) oder nach der Produkt- und Hersteller-ID aus ihrer udev-Regel benannt.

Beispiel für die Syntax in einer evmap:

```
KEY_PROG1       = KEY_MENU #Ich bin ein Kommentar
```

Hier wird das Tasten-Event **KEY_PROG1** umbenannt. Der VDR empfängt dadurch das Lirc-Event **KEY_MENU**

2.3.1.4 Informationen über den Empfänger sammeln

Einen ersten Überblick, ob der Empfänger erkannt wurde gibt die Auflistung der Eingabegeräte des des Systems:

```
cat /proc/bus/input/devices
```

Hier werden alle Eingabegeräte aufgelistet, für die ein Eintrag unter */dev/input/* angelegt wird. Für USB-Lirc-Empfänger, die automatisch konfiguriert werden, oder für über das Webfrontend eingerichtete Lirc-Geräte gibt es keinen gesonderten Eintrag, sie können an der Zeile **N: Name="lircd"** erkannt werden.

Im folgenden mal Beispiele für verschiedene Eingabegeräte:

```
user@yaVDR:~$ cat /proc/bus/input/devices
```

Power-Button am Gehäuse

```
I: Bus=0019 Vendor=0000 Product=0001 Version=0000
N: Name="Power Button"
P: Phys=PNP0C0C/button/input0
S: Sysfs=/devices/LNXSYSTM:00/device:00/PNP0C0C:00/input/input0
U: Uniq=
H: Handlers=kbd event0
B: PROP=0
B: EV=3
B: KEY=10000000000000 0

I: Bus=0019 Vendor=0000 Product=0001 Version=0000
N: Name="Power Button"
P: Phys=LNXPWRBN/button/input0
S: Sysfs=/devices/LNXSYSTM:00/LNXPWRBN:00/input/input1
U: Uniq=
H: Handlers=kbd event1
B: PROP=0
B: EV=3
B: KEY=10000000000000 0
```

[10] Eventmaps

Angeschlossene Tastatur

```
I: Bus=0011 Vendor=0001 Product=0001 Version=ab41
N: Name="AT Translated Set 2 keyboard"
P: Phys=isa0060/serio0/input0
S: Sysfs=/devices/platform/i8042/serio0/input/input2
U: Uniq=
H: Handlers=sysrq kbd event2
B: PROP=0
B: EV=120013
B: KEY=402000000 3803078f800d001 feffffdfffefffff fffffffffffffffe
B: MSC=10
B: LED=7
```

direkt eingebundene Fernbedienung, hier Hama MCE Remote mit Tastatur- und Mausfunktion

```
I: Bus=0003 Vendor=05a4 Product=9881 Version=0110 (1)
N: Name="HID 05a4:9881"
P: Phys=usb-0000:00:02.0-2/input0
S: Sysfs=/devices/pci0000:00/0000:00:02.0/usb2/2-2/2-2:1.0/input/input3
U: Uniq=
H: Handlers=sysrq kbd event3 (2)
B: PROP=0
B: EV=120013
B: KEY=e080ffdf01cfffff fffffffffffffffe
B: MSC=10
B: LED=7

I: Bus=0003 Vendor=05a4 Product=9881 Version=0110 (3)
N: Name="HID 05a4:9881"
P: Phys=usb-0000:00:02.0-2/input1
S: Sysfs=/devices/pci0000:00/0000:00:02.0/usb2/2-2/2-2:1.1/input/input4
U: Uniq=
H: Handlers=kbd mouse0 event4 (4)
B: PROP=0
B: EV=17
B: KEY=1f0000 2020000 3878d801d001 1e000000000000 0
B: REL=103
B: MSC=10
```

(1), (3) Die Tastenbelegung von direkt eingebundenen Eingabegeräten kann über Evmaps angepasst werden

(2), (4) Wollte man die direkt vom Eingabegerät ausgegebenen Tastendrücke beobachten, müsste man nach dem Stoppen von eventlircd das Tool evtest mit dem Pfad */etc/input/event3* bzw. */etc/input/event4* aufrufen.

Eventlircd

```
I: Bus=0003 Vendor=05a4 Product=9881 Version=0110
N: Name="eventlircd"
P: Phys=/dev/input/event4
S: Sysfs=/devices/virtual/input/input5
U: Uniq=
H: Handlers=mouse1 event5
B: PROP=0
B: EV=7
B: KEY=10000 0 0 0 0
B: REL=3
```

Dieses Eingabegerät existiert immer, wenn eventlircd läuft. Es ist das virtuelle Eingabegerät, das alle Tastendrücke der angeschlossenen und entsprechend Konfigurierten Empfänger sammelt und an den VDR weiterleitet.

rc-core Empfänger

```
I: Bus=0018 Vendor=0000 Product=0000 Version=0000
N: Name="i2c IR (Hauppauge WinTV PVR-350"
P: Phys=i2c-2/2-0018/ir0
S: Sysfs=/devices/virtual/rc/rc0/input6 (1)
U: Uniq=
H: Handlers=kbd event6
B: PROP=0
B: EV=100013
B: KEY=108fc010 2468d100000000 0 18000 18040002801 8e168000000000 ffe
B: MSC=10
```

(1) Existiert ein Gerät mit einem Sysfs-Pfad, der mit **/devices/virtual/rc/rc** beginnt, kann es mittels ir-keytable konfiguriert werden.

Lirc-Gerät mit --uinput

```
I: Bus=0000 Vendor=0000 Product=0000 Version=0000
N: Name="lircd"
P: Phys=
S: Sysfs=/devices/virtual/input/input7
U: Uniq=
H: Handlers=sysrq rfkill kbd event7
B: PROP=0
B: EV=100003
B: KEY=ffffffffffffffff ffffffffffffffff ffffffffffffffff ffffffffffffffff ffffffffffffffff ↩
   ffffffffffffffff ffffffffffffffff ffffffffffffffff ffffffffffffffff ffffffffffffffff ↩
   ffffffffffffffff fffffffffffffffe
```

2.3.2 ir-keytable

ir-keytable gibt Auskunft über das angelegte Inputdevice(*/dev/input/eventX*), das verwendete Protokoll und den verwendeten Treiber für IR-Eingabegeräte, die **rc-core** bzw. das Treibermodul **ir-core** verwenden. Außerdem ist das Programm dafür zuständig Tastenbelegungen zu laden, die nicht dem vorgegebenen Stand in den rc-core Treibern entsprechen.



Anpassung der Tastenbelegung für rc-core Empfänger
Im Gegensatz zu Empfängern, die als Tastatur erkannt werden, deren Tastenbelegung nur mittels evmaps geändert werden kann, macht es für rc-core Empfänger keinen Sinn die Tastenbelegung über evmaps zu ändern. Sinnvoller ist es eine angepasste Keytable zu erstellen und diese über eine Regel in der */etc/rc_maps.cfg* beim Start automatisch laden zu lassen.
Falls bereits alle Tasten eine Ausgabe mit definiertem Namen liefern, hilft das hier gepostete Skript eine benutzerdefinierte Keytable zu erstellen und laden zu lassen, deren Tastennamen und -belegung man dann sicher vor eventuellen Updates der bei yaVDR mitgelieferten Keytables anpassen kann.

Beispiel für eine Hauppauge PVR350 mit IR-Empfänger:

```
user@yavdr:~$ sudo ir-keytable
Found /sys/class/rc/rc0/ (/dev/input/event3) with:
```



```
        Driver ir-kbd-i2c, table rc-rc5-tv                    ❷
        Supported protocols: RC-5                             ❸
        Enabled protocols:
Repeat delay = 500 ms, repeat period = 33 ms
```

❶ Die Fernbedienung wurde unter */dev/input/event3* eingebunden

❷ Der verwendete Treiber ist **ir-kbd-i2c**, die von ir-core geladene Keytable heißt rc-rc5-tv. Mit diesen Informationen

❸ Das unterstützte Protokoll ist **RC-5**

2.3.2.1 geladene Keymap(s) auslesen

Der folgende Befehl gibt die bereits für einen Empfänger geladene Zuordnung von Scancodes und Tastennamen aus. Das ist insbesondere dann hilfreich, wenn zwar eine Keymap geladen wurde, aber die Benennung der Tasten noch nicht Namespace-konform ist.

```
sudo ir-keytable -r
```

2.3.2.2 Scancodes ermitteln

Mit **ir-keytable** kann man auch Tastencodes der Fernbedienung ermitteln, z.B. um zu sehen ob überhaupt Tastendrücke erkannt werden oder um eine Keymap zu erstellen, die zur */vdr/lib/vdr/remote.conf* des yaVDR passt. Da Tastendrücke von eventlircd abgefangen werden, muss dieser Dienst gestoppt sein. Außerdem ist es empfehlenswert den VDR vorher zu stoppen, um eine unbeabsichtigte Fehlbedienung zu vermeiden.

Beispielausgabe ohne geladene Keytable

```
user@yavdr:~$ sudo ir-keytable -t
Testing events. Please, press CTRL-C to abort.
1310546299.271540: event MSC: scancode = 3d
1310546300.071543: event MSC: scancode = 3b
1310546301.471538: event MSC: scancode = 01
1310546301.871547: event MSC: scancode = 02
1310546302.271537: event MSC: scancode = 03
```

2.3.2.3 Keymap erstellen

Passend zur eigenen Fernbedienung kann man sich nun eine Keymap erstellen, dabei muss der scancode in eine Hexadezimalzahl übertragen und der Taste eine Funktion zugewiesen werden. Aus **scancode = 3d** wird also **0x003d**, aus **scancode = 01** wird **0x0001** usw.

Die Keymap wird dann am besten unter einem möglichst aussagekräftigen Namen unter */etc/rc_keymaps/* gespeichert, wo sie keine Gefahr läuft von mit yaVDR-Updates ausgelieferten Keytables überschrieben zu werden. Die von yaVDR mitgelieferten Keytables liegen unter */lib/udev/rc_keymaps/* und können als Vorlage für angepasste Keytables verwendet werden.

Beispiel für eine selbst definierte Keytable, die unter /etc/rc_keymaps/pvr350 gespeichert würde (Auszug)

```
# table rc-rc5-tv, type: RC-5 ❶
0x003b KEY_SETUP
0x003d KEY_POWER2
0x0001 KEY_1
0x0002 KEY_2
0x0003 KEY_3
[...]
```

❶ Die erste Zeile der Keytable sollte den Namen der Keytable sowie den verwendeten Protokoll-Typ (hier RC-5) enthalten.



Kommentare in den Keymaps

ir-keytable akzeptiert keine reinen Kommentarzeilen. Wird eine Kopfzeile verwendet, die Keytable-Name und Fernbedienungsprotokoll angibt, so muss darauf geachtet werden, dass für beide Parameter erlaubte Werte eingetragen sind.

Falsch wäre also:

```
# table rc-rc5-tv, type  ❶
# Ich bin ein Kommentar  ❷
0x003b KEY_SETUP
0x003d KEY_POWER2
0x0001 KEY_1
0x0002 KEY_2
0x0003 KEY_3
[...]
```

❶ Kein Protokolltyp angegeben

❷ reine Kommentarzeilen sind nicht erlaubt

Korrekt geht es so:

```
# table rc-rc5-tv, type: RC-5 ❶
0x003b KEY_SETUP # Ich bin ein Kommentar ❷
0x003d KEY_POWER2
0x0001 KEY_1
0x0002 KEY_2
0x0003 KEY_3
```

❶ Vollständige Angabe des Namens der Keytable und des verwendeten Protokolls

❷ Kommentare hinter der Tastendefinition sind erlaubt

2.3.2.4 Keymap laden

Damit die Keytable auch automatisch geladen wird, muss eine Regel in */etc/rc_maps.cfg* für die Fernbedienung erstellt werden. Da die Datei getemplated ist, sollte man ein entsprechendes Custom-Template für die eigene Einträge erstellen, um ein Überschreiben der Regel durch Updates von yaVDR zu vermeiden.

/etc/rc_maps.cfg

```
# driver          table                          file
❶ ir-kbd-i2c       rc-rc5-tv                      /etc/rc_keymaps/pvr350
```

1 Den Treiber und den Namen der Keytable haben wir wie weiter oben beschrieben mittels sudo ir-keytable ermittelt.

Zu Testzwecken kann man Einstellungen für per ir-core eingebundene Geräte auch von Hand laden oder zurücksetzen.

Wie wird die rc_maps.cfg ausgewertet?
Dazu wird eine udev-Regel in /lib/udev/rules.d/40-ir-keytable.rules benutzt, die gemäß den Regeln in der /etc/rc_maps.cfg die dort eingestellten Keymaps lädt (ir-keytable -a /etc/rc_maps.cfg).

Manuelles Laden einer Keymap durch ir-keytable

```
user@yavdr:~$ sudo ir-keytable -c ❶ -p rc-5 ❷ -w /etc/rc_keymaps/pvr350 ❸
```

1 **-c** entfernt alle bereits geladenen Keymaps

2 **-p <Protokoll>** setzt das IR-Protokoll

3 **-w <keymap>** gibt die zu ladende Keymap an. Weitere Informationen findet man in der Manpage von **ir-keytable**.

Beispielausgabe von ir-keytable -t für die gleichen Tasten mit passender Keytable

```
user@yavdr:~$ sudo ir-keytable -t
Testing events. Please, press CTRL-C to abort.
1310546400.571539: event MSC: scancode = 3d
1310546400.571545: event key down: KEY_POWER2 (0x0164)
1310546400.571547: event sync
1310546400.820703: event key up: KEY_POWER2 (0x0164)
1310546400.820705: event sync
1310546401.471543: event MSC: scancode = 3b
1310546401.471551: event key down: KEY_SETUP (0x008d)
1310546401.471553: event sync
1310546401.720700: event key up: KEY_SETUP (0x008d)
1310546401.720703: event sync
1310546406.271547: event MSC: scancode = 01
1310546406.271556: event key down: KEY_1 (0x0002)
1310546406.271558: event sync
1310546406.520700: event key up: KEY_1 (0x0002)
1310546406.520703: event sync
1310546407.071539: event MSC: scancode = 02
1310546407.071543: event key down: KEY_2 (0x0003)
1310546407.071545: event sync
1310546407.320705: event key up: KEY_2 (0x0003)
1310546407.320709: event sync
1310546407.871539: event MSC: scancode = 03
1310546407.871544: event key down: KEY_3 (0x0004)
1310546407.871546: event sync
1310546408.120700: event key up: KEY_3 (0x0004)
1310546408.120702: event sync
```

2.3.3 evtest

evtest ist ein Programm, das es erlaubt nähere Informationen über ein unter */dev/input/event<X>* eingebundenes Gerät zu erhalten. Es kann über das Paket **evtest** nachinstalliert werden. Um eine Ausgabe von Tastendrücken sehen zu können, muss man eventlircd zuvor stoppen.

```
user@yavdr:~$ sudo evtest /dev/input/event3
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0    ❶
Input device name: "i2c IR (Hauppauge WinTV PVR-350"           ❷
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
    Event code 1 (Esc)                                           ❸
    Event code 2 (1)
    Event code 3 (2)
    Event code 4 (3)
    Event code 5 (4)
    Event code 6 (5)
    Event code 7 (6)
    Event code 8 (7)
    Event code 9 (8)
    Event code 10 (9)
    Event code 11 (0)
    Event code 103 (Up)
    Event code 105 (Left)
    Event code 106 (Right)
    Event code 108 (Down)
    Event code 113 (Mute)
    Event code 114 (VolumeDown)
    Event code 115 (VolumeUp)
    Event code 119 (Pause)
    Event code 128 (Stop)
    Event code 139 (Menu)
    Event code 141 (Setup)
    Event code 158 (Back)
    Event code 167 (Record)
    Event code 168 (Rewind)
    Event code 207 (Play)
    Event code 208 (Fast Forward)
    Event code 352 (Ok)
    Event code 356 (Power2)
    Event code 358 (Info)
    Event code 359 (Time)
    Event code 363 (Channel)
    Event code 365 (EPG)
    Event code 366 (PVR)
    Event code 370 (Subtitle)
    Event code 373 (Mode)
    Event code 388 (Text)
    Event code 398 (Red)
    Event code 399 (Green)
    Event code 400 (Yellow)
    Event code 401 (Blue)
    Event code 402 (ChannelUp)
    Event code 403 (ChannelDown)
    Event code 407 (Next)
    Event code 412 (Previous)
  Event type 4 (Misc)
    Event code 4 (ScanCode)
  Event type 20 (Repeat)
Testing ... (interrupt to exit)
```

❶ Produkt- und Hersteller-ID

❷ Produktname, z.B. für eine udev-Regel

(3) Ab hier sind die verfügbaren Tasten der Fernbedienung aufgelistet

2.3.4 udev

udev ist ein Dienst, der es erlaubt Hardware dynamisch zu erkennen und in das System einzubinden. Für jedes Gerät, das vom Hersteller i.d.R. mit individuellen Identifikationsmerkmalen ausgestattet wurde, kann so anhand von udev-Regeln festgelegt werden, welche Treiber, Module oder Befehle aktiviert werden sollen, sobald ein Gerät erkannt wird.

2.3.4.1 udev-Regeln

Über Regeln in der Datei */lib/udev/rules.d/98-eventlircd.rules* kann festgelegt werden, welche Konfiguration eventlircd für angeschlossene USB-Fernbedienungen lädt.

Beispiel für einen Eintrag in /lib/udev/rules.d/98-eventlircd.rules

```
ENV{ID_VENDOR_ID}=="0419", ENV{ID_MODEL_ID}=="0001", \                    (1)
  ENV{eventlircd_enable}="true",\                                         (2)
  ENV{eventlircd_evmap}="03_$env{ID_VENDOR_ID}_$env{ID_MODEL_ID}.evmap"   (3)
```

(1) Die Fernbedienung wird anhand der ID_VENDOR_ID und der ID_MODEL_ID erkannt (siehe evtest)

(2) Erlaubt eventlircd die Eingaben des Geräts zu verwerten

(3) Lädt die evmap für das Gerät. In diesem Fall entsteht nach Einsetzen der Umgebungsvariablen der Name "03_0419_0001.evmap", demnach wird */etc/eventlircd.d/03_0419_0001.evmap* als Keymap geladen.



Sonderfälle
In einigen Fällen kann mittels Vendor-ID und Model-ID nicht zuverlässig zwischen bestimmten Empfängern unterschieden werden (z.B. X10-Empfänger). Hier werden weitere udev-Attribute ausgewertet. Die für diese Empfänger angepassten udev-Regeln sind unter */lib/udev/rules.d/98-eventlircd-names.conf* zu finden.

2.3.4.2 udev-Crashkurs

TODO:

- näheres zu UDEV, mitgelieferte und benutzerdefinierte Regeln, UDEV-Merkmale ermitteln <<

TODO

- udev für USB-FB

Weitere Informationen über ein Eingabegerät kann man mittels udevadm herausfinden. Es eignet sich gut, wenn man Kriterien für udev-Regeln bestimmen möchte:

udevadm

```
user@yavdr:~$ sudo udevadm info --query=all --attribute-walk --name=/dev/input/event4

Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

  looking at device '/devices/virtual/input/input4/event4':
    KERNEL=="event4"
    SUBSYSTEM=="input"
    DRIVER==""

  looking at parent device '/devices/virtual/input/input4':
    KERNELS=="input4"
    SUBSYSTEMS=="input"
    DRIVERS==""
    ATTRS{name}=="Sundtek Ltd. Remote Control"
    ATTRS{phys}==""
    ATTRS{uniq}==""
    ATTRS{modalias}=="input:b0003v0000p0000e0001-e0,1,2,4,14,k71,72,73,74,80,9E,A7,AE,CF ↩
        ,161,174,18E,18F,190,191,192,193,ramlsfw"
```

2.3.5 irw

Wenn eventlircd gestartet ist, zeigt der Befehl **sudo irw** die vom VDR erkannten Tastendrücke an. Die Ausgabe der erkannten Tastendrücke kann per **STRG + c** abgebrochen werden. **irw** kann auch auf den Sockets von lircd lauschen, wenn man ihm den Pfad zu diesem als Argument übergibt.

irw bei eventlircd mit einer Hama MCE Remote

```
user@yavdr:~$ sudo irw
(1) (2) (3)      (4)
67 0 KEY_UP   devinput
6c 0 KEY_DOWN devinput
```

(1) Tastencode

(2) Tastendruck 0=Event

(3) Name der Taste

(4) Die Tasten werden per devinput an eventlirc übergeben

irw für lircd am Beispiel eines Atric-Empfängers http://www.atric.de/IR-Einschalter/index.php mit Hauppauge A415-HPG-WE Fernbedienung

```
# PID von lircd bestimmen
user@yavdr:~$ sudo pidof lircd
765
user@yavdr:~$ sudo irw /var/run/lirc/lircd.765
(1)              (2) (3)      (4)
0000000000001794 00 KEY_UP hvr4000-2
0000000000001794 01 KEY_UP hvr4000-2
0000000000001795 00 KEY_UP hvr4000-2
0000000000001795 01 KEY_UP hvr4000-2

# Falls man sicher ist, dass nur eine lircd-Instanz läuft geht auch dieser Befehl:
sudo irw /var/run/lirc/lircd.$(pidof lircd)
```

1. Tastencode
2. 00=Taste gedrückt 01=Taste losgelassen
3. Name der Taste
4. Benutzte Fernbedienung aus der lircd.conf für die Tastenzuordnung

2.3.6 LIRC Empfänger

2.3.6.1 Serielle Empfänger

Serielle Empfänger wie z.B. der Atric-Einschalter oder Homebrew-Empfänger lassen sich leider schlecht automatisch vorkonfigurieren, da sie mit beliebigen Fernbedienungen benutzt werden können. Trotzdem lassen sie sich in yaVDR mit wenig Aufwand zusammen mit LIRC einrichten.

Einstellungen im Webfrontend Im Abschnitt Fernbedienung des yaVDR-Webfrontend setzt man das Häkchen bei Lirc-Unterstützung und wählt den geeigneten Empfänger aus der Liste. Für Atric-Einschalter ist dies der Eintrag "**Home-brew (16x50 UART compatible serial port)**". Außerdem muss noch die Serielle Schnittstelle angegeben werden, über die der Empfänger angeschlossen ist. Die Nummerierung orientiert sich normalerweise an den Angaben im Handbuch des Mainboards.

Anpassen der lircd.conf Im nächsten Schritt muss dann noch eine passende lircd.conf nach */etc/-lirc/lircd.conf* kopiert werden. Die dort verwendeten Tastennamen sollten sich an der remote.conf orientieren, da die Fernbedienung dann sowohl mit dem VDR als auch unter XBMC ohne weitere Konfiguration verwendet werden kann.

yaVDR 0.3 bzw. "alte" lircd.conf

```
[...]
begin codes
          Up                     0x1794
          Down                   0x1795
          Ok                     0x17A5
          left                   0x1796
          right                  0x1797
          info                   0x179B
[...]
```

yaVDR 0.4

```
[...]
begin codes
          KEY_UP                     0x1794
          KEY_DOWN                   0x1795
          KEY_OK                     0x17A5
          KEY_LEFT                   0x1796
          KEY_RIGHT                  0x1797
          KEY_INFO                   0x179B
[...]
```

Nach einem Neustart von eventlircd kann man die für den seriellen Empfänger konfigurierte Fernbedienung benutzen. Die Tasten-Ausgabe von lircd kann unter */var/run/lirc/lircd.<PID>* mittels irw ausgelesen werden.

2.3.6.2 Igor USB

Für die Igor USB-Empfänger muss lediglich eine passende Namespace-konforme Lircd.conf nach /etc/lirc/ kopiert werden. Über eine udev-Regel wird automatisch ein passender lircd-Prozess gestartet und dessen Ausgabe von eventlircd ausgewertet. Der Socket von lircd liegt typischerweise unter */var/run/lirc/lircd-lirc0* und kann mittels irw auf Funktion überprüft werden.



Cave Einstellungen zur Fernbedienung im Webfrontend!
Bei Igor USB-Empfängern darf Lirc nicht für den Igor über das Webfrontend aktiviert und eingerichtet werden, da sonst zwei lircd-Prozesse gestartet werden, die sich gegenseitig blockieren und die Funktion der Fernbedienung verhindern!

2.3.7 X10-Fernbedienungen

Müssen bei X10-Fernbedienungen Tastend doppelt gedrückt werden, damit der VDR reagiert, so muss noch die Modul-Option "keychange=1" für das Modul ati_remote gesetzt werden:

/etc/modprobe.d/ati_remote.conf

```
options ati_remote keychange=1
```

Keymapping für X10-Empfänger
Die Tastennamen für X10-Empfäger können in den evmaps */etc/eventlircd.d/03_0bc7_0006.evmap* bzw. */etc/eventlircd.d/03_0bc7_0006.pollin.evmap* angepasst werden. Die udev-Regeln, die für die Aktivierung der jeweiligen Keymap zuständig sind stehen in der */lib/udev/rules.d/98-eventlircd-names.conf*:
/lib/udev/rules.d/98-eventlircd-names.conf

```
ATTRS{name}=="X10 Wireless Technology Inc USB Receiver", \
  ENV{eventlircd_enable}="true", \
  ENV{eventlircd_evmap}="03_0bc7_0006.evmap"

ATTRS{name}=="X10 WTI RF receiver", \
  ENV{eventlircd_enable}="true", \
  ENV{eventlircd_evmap}="03_0bc7_0006_pollin.evmap"
```



Änderung durch linux-media-dkms
Nach Installation des Paketes **linux-media-dkms** werden die X10-Empfänger nicht mehr als tastaturartige Eingabegeräte erkannt, sondern durch ein neues Treibermodul (das ebenfalls den Namen **ati_remote** trägt) als rc-core Gerät eingebunden. Dementsprechend sollte die Anpassung der Tastennamen dann über eigene Keytables erfolgen.

2.4 Das Template-System

Unter yaVDR werden viele Konfigurationsdateien automatisch aus Vorlagen zusammengesetzt. Diese Vorlagen werden **Templates** genannt.

Die Templates, die das yaVDR-Team mit seinen Paketen ausliefert, sind so genannte **Standard-Templates**. Eine Übersicht der Standard-Templates befindet sich im Entwicklerbereich auf der yaVDR Homepage.

Wollte man nun etwas direkt an einer Konfigurationsdatei ändern, die aus Templates generiert wird, entstünde folgendes Problem: Die manuelle Änderung würde durch Skripte beim Ändern von Einstellungen über das Webfrontend oder beim nächsten Update eines Paketes überschrieben, da dabei die Konfigurationsdateien aus den Standard-Templates neu generiert werden würden.

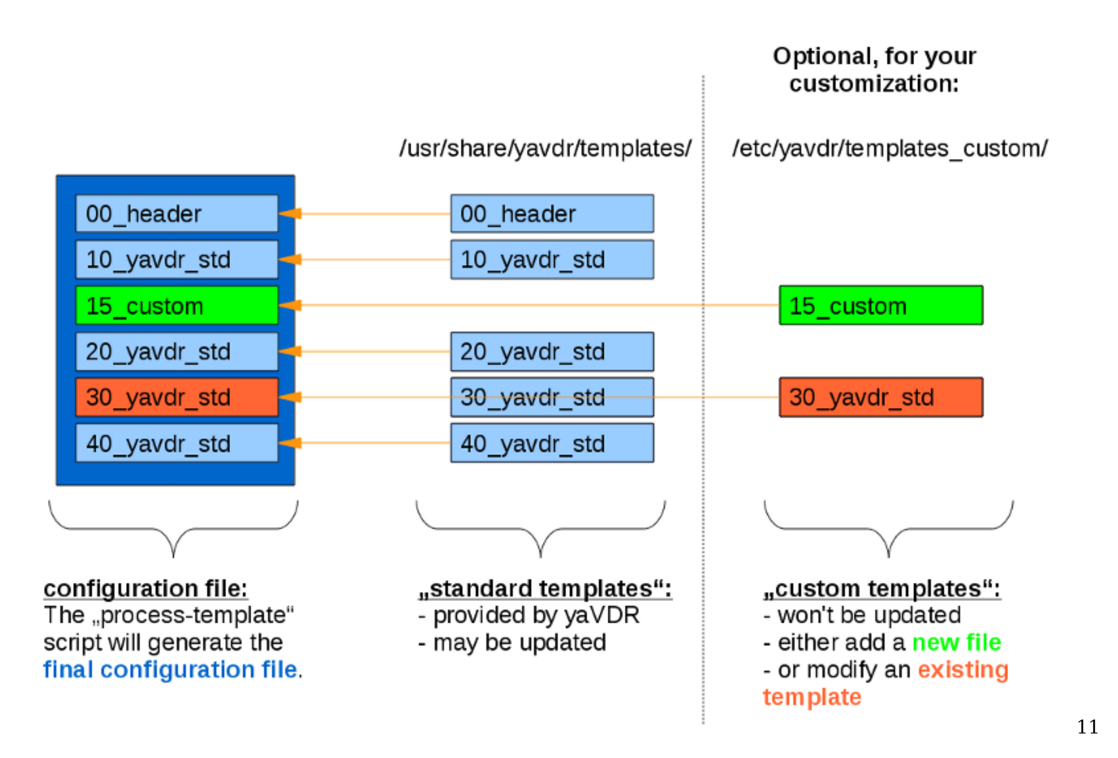
Änderungen an den Standard-Templates wären deshalb gleichermaßen nicht sinnvoll, da diese bei einem Paketupdate ebenfalls aktualisiert werden würden und manuelle Änderungen dabei verloren gingen.

Um dennoch Veränderungen an Konfigurationsdateien vorzunehmen wurde die Möglichkeit geschaffen benutzerdefinierte Vorlagen, so genannte **Custom-Templates** anzulegen, welche die Standard-Templates ergänzen oder Teile von ihnen überschreiben.

Alle **Custom-Templates** müssen unterhalb des Ordners */etc/yavdr/templates_custom/* abgelegt werden. Benutzerdefinierte Vorlagen in diesem Ordner werden bei Paketupdates nicht verändert.

Wo befinden sich die Templates?
Standard-Templates liegen unter */usr/share/yavdr/templates/*
Custom-Templates sind unter */etc/yavdr/templates_custom/* zu finden.
Zusätzliche Custom-Templates werden in Konfigurationsdateien eingefügt, gleichnamige Custom-Templates überschreiben das entsprechende Standard-Template.
Man kann die Erstellung einer Konfigurationsdatei aus den Templates so aufrufen:

```
user@yavdr04:~$ sudo process-template <Pfad zur Konfigurationsdatei>
```



[11]

[11] http://www.vdr-wiki.de/wiki/index.php/Datei:Yavdr_templating-visualization.png

2.4.1 Beispiel anhand der Datei */etc/vdr/vdr-addon-acpiwakeup.conf*

Anhand der Konfigurationsdatei für vdr-addon-acpiwakeup soll das Template-System an einem Beispiel verdeutlicht werden. Wir möchten, dass der VDR jeden Tag um 19:45 aufwacht, damit das EPG aktualisiert und der VDR die Gelegenheit hat nach neuen Treffern für die Suchtimer zu suchen. Außerdem müssen wir ihn dann nicht extra einschalten, wenn wir die Tagesschau sehen wollen.

Die Datei */etc/vdr/vdr-addon-acpiwakeup.conf* sieht im Auslieferungszustand in so aus [12]:

```
################################################################################
#                                                                              #
#   The following configuration file is generated automatically by the yaVDR    #
#     system. Don't change this file as every update of yaVDR will overwrite    #
#         the local changes. Instead put your required customizations          #
#       into /etc/yavdr/templates_custom/ based on the original templates       #
#                      under /usr/share/yavdr/templates.                       #
#                                                                              #
#            http://www.yavdr.org/developer-zone/template-overview/            #
#                                                                              #
#                                                                              #
################################################################################
# Activate/deactivate ACPIWakeup with yes/no:
ACPI_ENABLED="yes"

# How many minutes should the machine wake up before the timer starts:
ACPI_START_AHEAD=5

# If you want your VDR machine to wakeup in regular intervals (i.e. for
# updating EPG data), specify the days of the week and the wakeup time.
#
# Days of the week for regular wakeup (not set=Disabled, 1=Monday...7=Sunday)
# ACPI_REGULAR_DAYS="1 2 3 4 5 6 7"

# Wakeup time
#ACPI_REGULAR_TIME=01:00  # HH:MM
```

Sie wird aus den Standard-Templates in */usr/share/yavdr/templates/etc/vdr/vdr-addon-acpiwakeup.conf/* zusammengesetzt:

Beispiel für ein Template-Verzeichnis

```
user@yavdr04:~$  ls -1 /usr/share/yavdr/templates/etc/vdr/vdr-addon-acpiwakeup.conf/
02_header               ❶
10_enable-acpiwakeup    ❷
20_start-ahead          ❸
30_regular-days         ❹
```

1. Template für die Warnung, Templates zu verwenden
2. Template um ACPI-Wakeup erlauben
3. Template um Zeitspanne einzustellen, um rechtzeitig vor dem Timer aufzuwachen
4. Template der Einstellungen für regelmäßiges Aufwachen

Wir wollen nun ein Custom-Template für das Standard-Template *30_regular-days* erstellen.

Dazu erstellen wir einen entsprechenden Ordner in dem Verzeichnis für die Custom-Templates und kopieren das Standard-Template, dass wir anpassen wollen dorthin (da die anderen Templates nicht

[12] Wenn im Webfrontend acpi-wakeup aktiviert ist

geänderd werden sollen, benötigen wir sie nicht als Custom-Templates, und sie dürfen daher auch weiterhin durch zukünftige Updates überschrieben werden):

```
user@yavdr04:~$ sudo mkdir -p /etc/yavdr/templates_custom/etc/vdr/vdr-addon-acpiwakeup.conf
user@yavdr04:~$ sudo cp /usr/share/yavdr/templates/etc/vdr/vdr-addon-acpiwakeup.conf/30 ←
    _regular-days /etc/yavdr/templates_custom/etc/vdr/vdr-addon-acpiwakeup.conf/30_regular- ←
    days
```

Jetzt passen wir mit dem Text-Editor unserer Wahl dieses Custom-Template an und ändern es so ab, dass der VDR jeden Abend um 19:45 aufwacht:

```
user@yavdr04:~$ sudo nano /etc/yavdr/templates_custom/etc/vdr/vdr-addon-acpiwakeup.conf/30 ←
    _regular-days
```

```
# If you want your VDR machine to wakeup in regular intervals (i.e. for
# updating EPG data), specify the days of the week and the wakeup time.
#
# Days of the week for regular wakeup (not set=Disabled, 1=Monday...7=Sunday)
ACPI_REGULAR_DAYS="1 2 3 4 5 6 7"

# Wakeup time
ACPI_REGULAR_TIME=19:45  # HH:MM
```

Nachdem das Custom-Template gespeichert wurde, lassen wir die Konfigurationsdatei aus den Templates neu erstellen:

```
user@yavdr04:~$ sudo process-template /etc/vdr/vdr-addon-acpiwakeup.conf
```

Die neu erzeugte Datei sieht nun so aus:

```
################################################################################
#                                                                              #
#   The following configuration file is generated automatically by the yaVDR    #
#     system. Don't change this file as every update of yaVDR will overwrite    #
#         the local changes. Instead put your required customizations          #
#       into /etc/yavdr/templates_custom/ based on the original templates      #
#                       under /usr/share/yavdr/templates.                      #
#                                                                              #
#             http://www.yavdr.org/developer-zone/template-overview/           #
#                                                                              #
#                                                                              #
################################################################################
# Activate/deactivate ACPIWakeup with yes/no:
ACPI_ENABLED="yes"

# How many minutes should the machine wake up before the timer starts:
ACPI_START_AHEAD=5

# If you want your VDR machine to wakeup in regular intervals (i.e. for
# updating EPG data), specify the days of the week and the wakeup time.
#
# Days of the week for regular wakeup (not set=Disabled, 1=Monday...7=Sunday)
ACPI_REGULAR_DAYS="1 2 3 4 5 6 7"

# Wakeup time
ACPI_REGULAR_TIME=19:45  # HH:MM
```

2.5 Dienste von yaVDR

Damit yaVDR funktionieren kann, müssen einige Programme als Dienste im Hintergrund laufen, andere werden nur vorrübergehend benötigt, z.B. um Programme mit bestimmten Rechten aufzurufen.

TODO: Samba, Avahi-Mounter, Webfrontend

2.5.1 Upstart

Upstart wird von yaVDR für alle wichtigen Systemdienste und zum Start von Programmen verwendet. Dazu gehören vor allem der VDR selbst, aber auch der Dienst für die Fernbedienung, der Openbox und viele andere.

Eine Übersicht der von Upstart verwalteten Einträge kann man sich mit folgendem Kommando anzeigen lassen:

```
initctl list
```

Um Dienste zu steuern gibt es unter anderen folgende Kommandos, die wichtig sein können, wenn man von Hand Änderungen am System vornehmen möchte:

```
sudo start <Dienst>    ❶
sudo stop <Dienst>     ❷
sudo restart <Dienst>  ❸
sudo status <Dienst>   ❹
```

❶ Startet einen Dienst

❷ Stoppt einen Dienst

❸ Startet einen Dienst neu

❹ Zeigt den Status des Dienstes an (also ob er gerade aktiv ist oder nicht)

2.5.1.1 Beispiele

Im folgenden sind einige Beispiele für die Steuerung von Diensten aufgeführt. Sie lassen sich problemlos auf andere Dienste wie z.B. **openbox**, **vdr-frontend** oder **lircd** übertragen.

VDR starten

```
sudo start vdr
```

VDR stoppen

```
sudo stop vdr
```

VDR neu starten

```
sudo restart vdr
```

Status des VDR anzeigen

```
sudo status vdr
```

eventlircd starten

```
sudo start eventlircd
```

eventlircd stoppen

```
sudo stop eventlircd
```

eventlircd neu starten

```
sudo restart eventlircd
```

Status von eventlircd anzeigen

```
sudo status eventlircd
```

2.5.2 NFS

yaVDR bringt einen vorkonfigurierten NFShttp://de.wikipedia.org/wiki/Network_File_System[NFS]-Server mit, der die Medienverzeichnisse im Netzwerk bereitstellt. Sie werden zusäztzlich über Avahi announciert, so dass ein PC mit yaVDR 0.4 über den eingebauten avahi-mounter z.B. die Aufnahmeverzeichnisse aller anderen Rechner mit yaVDR 0.4 im Netzwerk automatisch finden und an entsprechender Stelle einbinden kann.

Auch Aufnahmen von NFS-Freigaben auf NAS oder anderen PCs lassen sich über den Avahi-Mounter einbinden. Dazu genügt es den entsprechenden Pfad zur Freigabe im Webfrontend einzutragen und abzuspeichern. Nach einem Neustart sollte das Verzeichnis dann ebenfalls berücksichtigt werden.

Weitere Verzeichnisse können über die */etc/exports* freigegeben werden. Im Ubuntuuser-Wiki findet sich ein einsteigerfreundlicher Artikel zu NFS

2.5.3 SAMBA

Samba bildet die von Windows bekannten Drucker- und Dateifreigaben nach. Aufgrund der guten Unterstützung dieses Protokolls in nahezu allen gängigen Betriebssystemen ist es gut geeignet um von Windows-, OS X- oder Linux-Systemen aus auf die Medienverzeichnisse des yaVDR übers Netzwerk zuzugreifen.

Die Konfigurationsdateien befinden sich unter */etc/samba/smb.conf*, die die voreingestellten Freigaben des yaVDR aus den folgenden Dateien bezieht (über "include" eingebunden):

- */etc/samba/smb.conf.yavdr.audio* für Audiodateien
- */etc/samba/smb.conf.yavdr.custom* für benutzerdefinierte Freigaben
- */etc/samba/smb.conf.yavdr.picture* für Bilder
- */etc/samba/smb.conf.yavdr.recordings* für Aufnahmen
- */etc/samba/smb.conf.yavdr.video* für Videos

Eigene Freigaben kann man z.B. in der *smb.conf.yavdr.custom* definieren. Wie ein solcher Eintrag aussehen muss und welche Optionen es dabei gibt, kann man in den Wiki-Artikelm zum Samba-Server und zur smb.conf auf http://www.ubuntuusers.de nachlesen.

2.6 Konfigurationsdateien und Schnittstellen

Das Webfrontend und das Einstellungsmenü des VDR sind in den meisten Fällen ausreichend, um den VDR zu konfigurieren. Für spezielle Anwendungsfälle oder zur Fehlerbehebung ist es aber gut zu wissen, wo man bei der manuellen Konfiguration ansetzen muss.

2.6.1 SVDRP

SVDRP [13] ermöglicht es über eine einfache TCP-Verbindung auf Port 6419 mit dem VDR zu kommunizieren und ihn zu steuern. Nähere Informationen findet man im Artikel zu SVDRP im VDR-Wiki. Die Kommunikation von Client und Server mittels SVDRP erfolgt unter anderem bei VDR-Plugins wie remotetimers, epgsync und streamdev-client - möchte man diese nutzen ist es notwendig, den Zugriff auf den Server wie im Folgenden beschrieben zu erlauben.

Zugriff von anderen Rechnern
Für den Zugriff über SVDRP von anderen Rechnern aus muss die Datei */etc/vdr/svdrphosts.conf* angepasst werden. Einige Beispiele sind im folgenden aufgeführt:

Beispieleinträge für /etc/vdr/svdrphosts.conf

```
127.0.0.1       (1)
192.168.1.115   (2)
192.168.1.0/24  (3)
192.168.1.0/16  (4)
0.0.0.0/0       (5)
```

(1) Dieser Eintrag darf nicht entfernt werden, da sonst der VDR mit bestimmten Plugins nicht mehr richtig funktioniert!

(2) Erlaubt dem Rechner mit der IP 192.168.1.115 den Zugriff

(3) Erlaubt allen Rechnern, deren IP mit 192.168.1. beginnt den Zugriff

(4), (5) Erlaubt allen Rechnern, deren IP mit 192.168. beginnt den Zugriff (entspricht Class C Subnetzen)

Erlaubt allen Rechnern unabhängig von ihrer IP den Zugriff (Achtung: damit ist ein ungeschützter Zugriff auf den VDR von außerhalb des eigenen Netzwerks möglich, falls der VDR nicht durch einen Router mit NAT [14] oder eine Firewall gesichert ist!)

2.7 Plugins

Der VDR selbst konzentriert sich auf die Kernfunktionen DVB-Empfang, das Aufzeichnen von Sendungen und eine rudimentäre EPG-Verwaltung. Zusätzliche Funktionen werden über Plugins realisiert. Eine gute Übersicht welche Plugins existieren und ob sie mit den unter yaVDR verwendeten VDR-Versionen 1.7.x kompatibel sind, bietet die Seite Plugins im VDR-Wiki. Die meisten kompatiblen Plugins sind in den yaVDR-Paketquellen zu finden und können bei Bedarf einfach über das Webfrontend oder die Shell nachinstalliert werden.

[13] Simple VDR Protocol
[14] Network Address Translation

2.7.1 Streamdev-Plugin

Das Streamdev Plugin erlaubt es dem VDR Live-TV an andere VDRs (über das Streamdev-Client Plugin) oder beliebige Clients via HTTP zu streamen. Es teilt sich in Streamdev-Server und Streamdev-Client auf.

2.7.1.1 Streamdev-Server

Das Streamdev-Server Plugin (Paketname **vdr-plugin-streamdev-server**) ist bei yaVDR 0.4 vorinstalliert. Es erweitert den VDR um die Fähigkeit Live-TV übers Netzwerk zu streamen.

Zugriff von anderen Rechnern aus
In der Voreinstellung nach der Installation erlaubt das Plugin nur Zugriffe vom gleichen Rechner. Um anderen Clients den Zugriff (und die Steuerung wichtiger Funktionen per SVDRP) zu erlauben müssen wie im folgenden beschrieben zwei Konfigurationsdateien angepasst werden

---

Für den Zugriff über SVDRP (z.B. wenn ein anderer VDR mit Streamdev-Client Plugin zugreifen können soll) muss die Datei */etc/vdr/svdrphosts.conf* wie unter SVDRP beschrieben angepasst werden.

Für den Zugriff auf die von Streamdev-Server bereitgestellten Transportströme muss außerdem die Datei */etv/vdr/plugins/streamdev-server/streamdevhosts.conf* angepasst werden. Diese Datei ist unter yaVDR 0.4 getemplated.

Beispieleinträge für /etc/vdr/plugins/streamdev-server/streamdevhosts.conf

```
127.0.0.1       (1)
192.168.1.115   (2)
192.168.1.0/24  (3)
192.168.0.0/16  (4)
0.0.0.0/0       (5)
```

(1) Dieser Eintrag darf nicht entfernt werden, da sonst der VDR mit bestimmten Plugins nicht mehr richtig funktioniert!

(2) Erlaubt dem Rechner mit der IP 192.168.1.115 den Zugriff

(3) Erlaubt allen Rechnern, mit der IP 192.168.1.xxx den Zugriff

(4) Erlaubt allen Rechnern in einem Class C Netzwerk (also mit IP 192.168.xxx.xxx, Standardeinstellung in yaVDR 0.4) den Zugriff auf den Streamdev-Server.

(5) Erlaubt allen Rechnern unabhängig von ihrer IP den Zugriff (Achtung: Nicht empfohlen, da damit ein ungeschützter Zugriff auf die Kanäle des VDR von außerhalb des eigenen Netzwerks möglich ist, falls der VDR nicht durch einen Router mit NAT oder eine Firewall gesichert ist!)

Der Streamdev-Server stellt auf Port 3000 eine Webinterface mit Links zu den von ihm angebotenen Streams bereit. Unter http://<IP_des_VDR>:3000/channels.m3u kann man eine Playlist mit allen Sendern herunterladen. Ein geeigneter Stream-Client ist z.B. der VLC Media Player.

Radio-Sender streamen
Möchte man Radio-Sender streamen, sollte man als Übertragungsformat unbedingt "ES" [a] wählen. Ein Beispiellink zum Stream sieht dann z.B. so aus: http://<IP_DES_VDR>:3000/ES/C-1-1093-28457

[a] **E**lementary **S**tream, siehe http://en.wikipedia.org/wiki/Elementary_stream

---

extremux.sh
Um die externremux.sh nutzen zu können, muss das Paket **mencoder** nachinstalliert werden.

2.7.1.2 Streamdev-Client

Das Streamdev-Client Plugin (Paketname: **vdr-plugin-streamdev-client**) erlaubt einem VDR auf das Streamdev-Server Plugin eines anderen VDR zuzugreifen und somit dessen DVB-Karten zu nutzen. Damit das klappt, müssen die gewünschten Kanäle aus der channels.conf des Servers in die channels.conf auf den Client kopiert werden und auf Server und Client die Verbindungseinstellungen angepasst werden. Es kann über eine Instanz des Streamdev-Clients immer nur ein Transponder übertragen werden.

Standardmäßig dürfen alle Clients in Class-C Netzwerken (IP-Bereicht 192.168.xxx.xxx) auf den Streamdev-Server zugreifen. Falls man den Zugriff genauer steuern oder auf andere Adressbereiche erweitern möchte, muss man dies in der */etc/vdr/plugins/streamdev-server/streamdevhosts.conf* entsprechend konfigurieren (siehe oben).

TODO

Konfigurationsoptionen des Plugins beschreiben, evtl. externremux.sh

2.7.2 xvdr-Plugin

Das xvdr-Plugin erlaubt experiementellen XBMC-Versionen mit PVR-Unterstützung Funktionen des VDR in von XBMC zu nutzen. Der Funktionsumfang umfasst aktuell Live-TV, EPG-Darstellung, das Verwalten von Timern sowie Zugriff auf die Aufnahmen des VDR. Einige Funktionen wie die Möglichkeit Schnittmarken zu benutzen und Aufnahmen zu schneiden sowie Einstellungen des VDR aus XBMC heraus zu verändern sind aktuell nicht vorhanden. Das xvdr-Plugin ist Multiuserfähig, d.h. es können mehrere XBMC-Clients auf einen VDR zugreifen und diesen nutzen.

2.7.2.1 Zugriffsrechte

Der Zugriff für von außen zugreifende Clients wird in der */var/lib/vdr/plugins/xvdr/allowed_hosts.conf* eingestellt. Die Syntax dieser Konfigurationsdatei entspricht der der svdrphosts.conf:

Zugriff von anderen Rechnern
Für den Zugriff von anderen Rechnern aus muss ggf. die Datei */var/lib/vdr/plugins/xvdr/allowed_hosts.conf* angepasst werden. Einige Beispiele sind im folgenden aufgeführt:

Beispieleinträge für /var/lib/vdr/plugins/xvdr/allowed_hosts.conf

```
127.0.0.1        (1)
192.168.1.115    (2)
192.168.1.0/24   (3)
192.168.1.0/16   (4)
0.0.0.0/0        (5)
```

(1) Dieser Eintrag sollte nicht entfernt werden, da sonst der lokale Zugriff von XBMC aus auf den VDR nicht mehr funktioniert!

(2) Erlaubt dem Rechner mit der IP 192.168.1.115 den Zugriff

3. Erlaubt allen Rechnern, deren IP mit 192.168.1. beginnt den Zugriff
4. Erlaubt allen Rechnern, deren IP mit 192.168. beginnt den Zugriff (entspricht Class C Subnetzen)
5. Erlaubt allen Rechnern unabhängig von ihrer IP den Zugriff (Achtung: damit ist ein ungeschützter Zugriff auf den VDR von außerhalb des eigenen Netzwerks möglich, falls der VDR nicht durch einen Router mit NAT [15] oder eine Firewall gesichert ist!)

2.7.3 dfatmo Plugin

Das Programm dfatmo und das dazugehörige Plugin für den VDR können über die Pakete **dfatmo** und **libxine-dfatmo-plugin** installiert werden.[16]

2.7.3.1 Konfiguration

Um das Plugin nutzen zu können, müssen die Start-Argumente von xine bzw. vdr-sxfe in der Konfigurationsdatei */etc/init/vdr-frontend.conf* angepasst werden. Diese Datei ist getemplated und sollte daher nicht direkt bearbeitet werden. Stattdessen muss der für das jeweilige Frontend passende Template-Teil (*20_xineliboutput* für **vdr-sxfe** und *25_xine* für **xine**) aus */usr/share/yavdr/templates/etc/init/vdr-frontend.conf/* nach */etc/yavdr/templates_custom/etc/init/vdr-frontend.conf/* kopiert und die Variable **XINELIBOUTPUTOPTS** bzw. **XINEOPTS** darin entsprechend angepasst werden. Zum Schluss wird die Konfigurationsdatei neu aus den Templates erzeugt.

Im Folgenden ist das ganze Schritt für Schritt beschrieben:

```
sudo mkdir -p /etc/yavdr/templates_custom/etc/init/vdr-frontend.conf/ ❶
sudo cp /usr/share/yavdr/templates/etc/init/vdr-frontend.conf/2* /etc/yavdr/ ↩
    templates_custom/etc/init/vdr-frontend.conf/ ❷
```

1. Erstellt den Ordner für benutzerdefinierte Template-Teile
2. Kopiert die nötigen Template-Teile

In den Dateien die Variable **XINE(LIBOUTPUT)OPTS** erweitern:

```
XINELIBOUTPUTOPTS="--post tvtime:method=use_vo_driver --reconnect --audio=alsa --syslog -- ↩
    silent --tcp"
XINELIBOUTPUTOPTS="$XINELIBOUTPUTOPTS --post=dfatmo:driver=serial,driver_param=/dev/ttyUSB0 ↩
    ,top=1,bottom=1,left=1,right=1,brightness=150,analyze_size=0,overscan=0,enabled=1"
```

bzw.

```
XINEOPTS="-G $GEOMETRY -A alsa --config /etc/xine/config --keymap=file:/etc/xine/keymap -- ↩
    post vdr --post vdr_video --post vdr_audio --verbose=2 --no-gui --no-logo --no-splash -- ↩
    deinterlace -pq"
XINEOPTS="$XINEOPTS --post=dfatmo:driver=serial,driver_param=/dev/ttyUSB0,top=1,bottom=1, ↩
    left=1,right=1,brightness=150,analyze_size=0,overscan=0,enabled=1"
```

Die Konfigurationsdatei aus den Templates neu erstellen:

```
process-template /etc/init/vdr-frontend.conf
```

Und zuletzt Openbox neu starten:

```
sudo stop openbox
sudo start openbox
```

[15] Network Address Translation
[16] http://www.die-keils.info/thomas/wordpress/?p=135

2.7.3.2 Ein/Ausschalten mit der Fernbedienung

Um das Atmolight per Fernbedienung schalten zu können, wurde die Taste KEY_BRIGHTNESS_CYCLE reserviert. Damit vdr-sxfe auf die Taste reagieren kann, muss es zwischen Eventlircd und VDR auf Tastendrücke lauschen. Die Variable **XINELIBOUTPUTOPTS** muss dann noch um die Hotkey-Unterstützung und den Socket von Eventlircd erweitert werden:

```
--hotkeys --lirc=/var/run/lirc/lircd
```

In der */etc/init/vdr.conf* muss das Start-Argument **--lircd=$LIRC** entfernt werden.

2.7.4 RestfulAPI-Plugin

yaVDR 0.4 bringt das neu entwickelte RestfulAPI-Plugin mit, das als Alternative (oder Ergänzung) zu SVDRP konzipiert ist. Es setzt nicht mehr auf direkte TCP Kommunikation sondern erlaubt die Interaktion mit dem VDR über HTTP-Anfragen. Die Antworten des VDR werden in den Formaten HTML, XML oder JSON ausgeliefert.

Vorteile:

- Die übertragung großer Datenmengen (z.B. EPG-Einträge eines Kanals) ist schneller
- Der VDR wird durch die Anfrage nicht blockiert, mehrere Clients stellen kein Problem dar
- Die ausgegebenen Daten lassen sich leichter verarbeiten
- Web-OSD für den VDR (noch in der Entwicklung) - damit kann jeder JS-fähige Browser das OSD des VDR darstellen
- Auch EPG-Bilder und Kanallogos lassen sich leicht vom VDR abrufen

Nachteile:

- Junges Plugin, das teilweise noch nicht den Funktionsumfang von SVDRP (gerade bei direkter Steuerung von Plugins) hat
- Noch keine große Zahl an Clients, die das Protokoll ausreizen

Die API des Plugins kann man unter folgender Webadresse einsehen: **http://<IP des VDR>:8002/info.html**

2.7.5 Dynamite-Plugin

Dieses Plugin erlaubt es DVB-Geräte dynamisch in den VDR einzubinden und wieder zu entfernen, ohne dass ein Neustart des VDR notwendig ist. Somit kann der VDR schneller starten, ohne z.B. auf "langsame" USB-TV-Karten zu warten. Sobald diese Initialisiert sind, werden sie basierend auf udev-Regeln nachträglich eingebunden und können genutzt werden. Für die Zukunft ist ein optionales automatisches Abschalten von gerade nicht benötigten DVB-Karten geplant um Strom zu sparen und die Lebensdauer der Komponenten zu verlängern.

2.8 Debugging

Wenn der VDR abstürzt ohne dass die Ursache aus dem syslog hervorgeht, kann man ein weitergehendes Debugging aktivieren um die Fehlerursache mittels Backtrace einzugrenzen.

Es müssen dazu die Pakete **vdr-dbg** und **gdb** installiert werden und die */etc/default/vdr* um einen Eintrag für vdr-dbg erweitert werden:

```
sudo su
echo "DAEMON=/usr/bin/vdr-dbg" >> /etc/default/vdr
restart vdr
exit
```

Beim nächsten Crash des VDR sollte in */var/log/vdr/* ein Backtrace des Segfaults erstellt werden.

Kapitel 3

Das kleine Linux 1x1

yaVDR basiert auf Ubuntu Linux. Einen Großteil der Konfiguration kann man über das **webinterface,Webinterface** vornehmen. Für komplexere Konfigurationen oder zur Fehlerbehebung ist es aber oft notwendig von Hand Änderungen vorzunehmen. Daher ist es ratsam sich mit der grundlegenden Bedienung eines Ubuntu-Systems über die Shell vertraut zu machen.

3.1 Rechteverwaltung unter Linux

TODO sudo Benutzer und Gruppen chown chmod

3.1.1 sudo - Super User

Unter Ubuntu ist standardmäßig kein Passwort für den root-Account vergeben. Um dennoch als root Befehle ausführen zu können gibt es das Kommando **sudo**, das Mitgliedern der Gruppe **admin** erlaubt Befehle mit erhöhten Rechten auszuführen.

3.2 Manpages

Möchte man mehr über einen Befehl oder ein Programm erfahren, hilft es zunächst die dazugehörige Manpage [1] dazu zu lesen. In ihr werden die Informationen zur Befehlssyntax, den Funktionen und Fähigkeiten eines Programms zusammengefasst. Standardmäßig werden die Manpages mit dem Programm **less** dargestellt. Innerhalb von **less** kann man mit den Pfeiltasten, PageUp & PageDown, der Leertaste usw. navigieren. [2]

Die Taste **q** beendet die Anzeige der Manpage.

Manpage von less

```
user@yavdr:$ man less
```

[1] also das Handbuch
[2] näheres dazu steht in der Manpage von less, siehe folgendes Beispiel ;)

3.3 Paketverwaltung

Ubuntu Linux verwendet das Paketverwaltungssystem APT, das es erlaubt Software-Pakete, die in den Repositories enthalten sind, zu verwalten, zu installieren und zu aktualisieren. Es gibt verschiedene Frontends für APT, im folgenden beschränkt sich diese Anleitung auf apt-get, ein weiteres Frontend, das sich für die Arbeit in der Shell eignet ist aptitude.

3.3.1 apt-get

apt-get ist das für yaVDR empfohlene Tool zur Paketverwaltung. Bevor man neue Pakete installiert, sollte man dafür sorgen, dass die Paketlisten auf dem aktuellen Stand und alle Update installiert sind, da es sonst zu Konflikten von Paketversionen oder nicht auflösbaren Abhängigkeiten kommen kann.

Paketlisten aktualisieren

```
sudo apt-get update
```

Alle Pakete auf den aktuellen Stand bringen

```
sudo apt-get dist-upgrade
```

Pakete installieren

```
sudo apt-get install Paketname                  (1)
sudo apt-get install Paket1 Paket2 Paket3       (2)
sudo apt-get install vim                        (3)
```

1. Installation von einem Paket
2. Installation von mehren Paketen
3. Beispiel für die Installation des Editors **vim**

Pakete suchen

```
sudo apt-cache search paketname
```

Informationen über ein Paket anzeigen

```
sudo apt-cache showpkg paketname
```

Pakete neu installieren

```
sudo apt-get install --reinstall Paketname1
```

Pakete entfernen

```
sudo apt-get remove Paketname    (1)
sudo apt-get purge Paketname     (2)
```

1. Paket entfernen, Konfigurationsdateien bleiben erhalten
2. Paket inklusive der Konfigurationsdateien entfernen

Nicht benötigte Pakete entfernen

--- sudo apt-get autoremove --- Dadurch werden alle überflüssigen Pakete entfernt, die aufgrund von Abhängigkeiten mitinstalliert wurden, momentan aber nicht gebraucht werden.

Lokal gespeicherte Pakete entfernen

--- sudo apt-get clean --- Entfernt alle lokal zwischengespeicherten Pakete, die aus den Paketquellen bezogen wurden.

3.3.2 Paketquellen

yaVDR nutzt im Auslieferungszustand die Ubuntu-Quellen sowie die stable-PPAs des yaVDR-Teams. Die aktuell eingetragenen Paketquellen finden sich unter */etc/apt/sources.list* und *etc/apt/sources.list.d/*.

Neue Paketquellen hinzufügen

Ausführliche Informationen zum Thema Paketquellen und PPA [3] findet man im Ubuntu-User Wiki. Möchte man neue Paketquellen hinzufügen, kann man diese entweder von Hand in die oben genannten Dateien eintragen (und den Schlüssel importieren) oder man benutzt für PPAs auf Launchpad den Befehl **add-apt-repository**:

PPAs hinzufügen

```
user@yavdr:~$ sudo add-apt-repository ppa:<Launchpad-Benutzer>/<Name des PPA>
```

3.4 Shell-Befehle vorzeitig beenden

Möchte man Befehle in der Shell vorzeitig beenden, kann man sie mit **STRG + c** abbrechen. Dies ist unter anderem bei Befehlen wie **irw** oder **tail -f** nützlich, da diese von sich aus unbegrenz lange laufen.

[3] Personal Package Archive

Kapitel 4

Häufig gestellte Fragen [1]

http://vdr-wiki.de/wiki/index.php/YaVDR/FAQ

Hier ensteht eine Sammlung der am häufigsten im http://www.vdr-portal.de gestellten Fragen. Sie wird mit der Zeit wachsen und hoffentlich zeitlich nah an den Stand der Entwicklung von yaVDR angepasst werden.

4.1 Fernbedienung

1. *Was bedeutet **Namespace-konform**?*

   In yaVDR dürfen aufgrund der Nutzung von Eventlircd nur bestimmte Tastennamen verwendet werden. Das bedeutet dass sie mit den Tastennamen in der Spalte "Event" in dieser Tabelle übereinstimmen müssen: https://bugs.yavdr.com/projects/1/wiki/IR_Integration_04#Eventmapping Wie man Tastennamen von noch nicht richtig vorkonfigurierten Fernbedienung anpassen kann, steht in der Hilfe zur Einrichtung der Fernbedienung, dem Abschnitt zu ir-keytable, Lirc und den Ev-maps.

2. *Ich habe einen Lirc-Empfänger und steige von yaVDR 0.3 auf 0.4 um - kann ich die Konfigurationsdateien einfach kopieren?*

   Nein. In yaVDR 0.4 sollte die remote.conf nicht mehr verändert werden. Die hardware.conf für Lircd wird über das Webfrontend erzeugt. Wichtig ist, dass die Tastennamen in der lircd.conf so angepasst werden, dass sie **Namespace-konform** sind.

3. *Wenn ich eine Taste auf der Fernbedienung drücke, werden doppelte Tastendrücke im VDR ausgeführt - wie kann ich das ändern?*

   Bis eine Lösung für eventlircd verfügbar ist, die die Anpassung des Repeat-Filters von Eventlircd erlaubt (siehe oneiric-Pakete im main PPA), bitte eines dieser eventlircd-Pakete verwenden und abstimmen wie groß der minimale Abstand zwischen zwei Tastendrücken sein muss: http://www.vdr-portal.de/board60-linux/board14-betriebssystem/board96-yavdr/108430-umfrage-zu-eventlircd-einstellung-des-repeat-filters/



Bug bei Verwendung von --uinput

Leider scheinen aktuelle Lirc-Versionen unter einem Bug bei der Übergabe von Tastendrücken bei Verwendung von **--uinput** zu leiden - es kommen dann doppelte Tastendrücke am Eventgerät an (die Tastendrücke an eigentlichen Lircd-Socket hingegen sind in Ordnung). Ein Workaround ist hier für über das WFE konfigurierte und hier für per udev-Regel eingebundene USB-Lirc-Empfänger im Forum beschrieben und wird voraussichtlich seinen Weg in die nächste yaVDR-Version finden.

---

[1] http://vdr-wiki.de/wiki/index.php/YaVDR/FAQ

Kapitel 5

Troubleshooting [1]

http://vdr-wiki.de/wiki/index.php/YaVDR/Troubleshooting

[1] http://vdr-wiki.de/wiki/index.php/YaVDR/Troubleshooting

Kapitel 6

Tipps und Tricks

6.1 Zugriffsrechte für Aufnahmen

Für bestimmte Anwendungsfälle ist es praktisch, wenn auch andere User Schreibrechte für die vom VDR erstellten Aufnahmen haben (z.B. weil man dann Projektdaten für Schnittprogramme direkt im Aufnahmeordner ablegen kann). Eine mögliche Lösung ist die fraglichen User der Gruppe **vdr** hinzuzufügen und die umask, mit der Dateien vom VDR erstellt werden anzupassen.

Hinzufügen eines Users zur Gruppe vdr

```
sudo adduser <username> vdr
```

umask für den VDR anpassen Folgende Zeile muss dann noch (am besten über ein custom Template) in die */etc/default/vdr* eingefügt werden:

```
umask 002
```

6.2 Samba und Windows

6.2.1 Ordnernamen im Aufnahmeverzeichnis

Unter Windows sind bestimmte Zeichen in Dateinamen nicht zulässig (z.B. "!",":" usw.) - diese tauchen allerdings gerne mal in Sendungstiteln auf und so werden diese dann beim Zugriff auf die per SAMBA freigegebenen Aufnahmeverzeichnisse des VDR nicht korrekt angezeigt. Um die Verwendung der problematischen Zeichen in den Ordnernamen zu unterbinden kann man die VDR mit der Option **--vfat** starten. Dazu muss (am besten per custom Template) die Zeile für die VDR-Optionen in der */etc/default/vdr* angepasst werden:

```
OPTIONS="-w 0 --vfat"
```

6.2.2 Symbolische Links

Verwendet man mehrere Aufnahmeverzeichnisse für den VDR (z.B. zusätzlich /srv/vdr/video.01 usw.) oder bindet Ordner selbt über symbolische Links ein, muss die */etc/samba/smb.conf* (am besten per custom Template) um drei Optionen erweitert werden, damit man auch über Samba-Freigaben darauf zugreifen kann:

```
# Follow Symlinks
   unix extensions = no
   follow symlinks= yes
   wide links= yes
```

6.3 Anpassen der Anthra-Skins

Die Anthra-Skins bieten einige Darstellungsoptionen, die nicht über das text2skin-Einstellungsmenü erreichbar sind. Um sie zu nutzen ist es notwendig die Skin-Datei zu bearbeiten oder das Paket mit den veränderten Einstellungen neu zu kompilieren.

Der generelle Ablauf zum Rekompilieren eines Pakets ist dabei folgendermaßen:

Als erstes müssen die zum Bauen benötigten Pakete installiert werden. Dies geschieht über **apt-get build-dep**

```
sudo apt-get build-dep <paketname>

# Beispiel für vdr-skin-anthra-1920fse:
sudo apt-get build-dep vdr-skin-anthra-1920-fse
```

Die Quelldateien selbst kann man sich mittels **apt-get source** in das Verzeichnis, in dem man sich gerade befindet herunterladen:

```
sudo apt-get source vdr-skin-anthra-1920-fse
```

Nun wechselt man in das Quellverzeichnis und passt die **Make.config** den eigenen Wünschen entsprechend an.

```
cd vdr-skin-anthra-1920-fse*
nano Make.config
```

Damit das zu erstellende Paket nicht beim nächsten Update überschrieben wird, erhöhen wir seine Versionsnummer.

```
dch -i
```

Beim ersten Aufruf kann der gewünschte Editor zum Bearbeiten der Datei gewählt werden. Hier am besten kurz die getätigten Änderungen in der Zeile mit dem Sternchen beschreiben und ggf. den Paketnamen in der obersten Zeile anpassen.

Nun wird das Paket gebaut:

```
dpkg-buildpackage -tc -us # Falls man einen gültigen LaunchPad-Account hat und dort einen ↩
    Schlüssel zur Signierung des Paketes hinterlegt hat, kann man das Paket natürlich auch ↩
    signieren (das Argument *-uc* entfällt dann).
```

Das erstellte Paket liegt nun im übergeordneten Ordner. Es kann mittels **dpkg -i** installiert werden:

```
cd ..
sudo dpkg -i vdr-skin-anthra-1920-fse_<Versionsnummer>_all.deb
```